

Form Approved  
OMB No. 0704-0188

It is estimated to average 100 per response, including the time for reviewing instructions, searching existing data sources, gathering and reviewing the pertinent information, and comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden. Write to Washington Headquarters, Bureau of Economic Analysis, Department of Commerce, Washington, DC 20540-0001, for a copy of OMB's (1217-0188) "Waste not, Want not."

DTIC  
ELECTE  
AUG 31 1992

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18  
298-102

Abstract of Dissertation Presented to the Graduate  
School of the University of Florida in Partial Fulfillment  
of the Requirements for the Degree of Doctor of Philosophy

CONFORMAL STEREOTACTIC RADIOSURGERY  
WITH MULTILEAF COLLIMATION

by

Dale C. Moss  
Captain, USAF, BSC

1992

352 pages

This paper outlines and implements a method of producing dose distributions that conform to any arbitrary, irregularly shaped target by means of dynamic conformal collimation using a multileaf collimator. The method may be summarized in three steps: production of the treatment plan; localization of the target; and the calculation of three dimensional dosimetry. Provision is made for volumetric evaluation of dosimetry with dose volume histograms and complication probability functions.

Treatment plans follow the standard of non-conformal stereotactic radiosurgery with several non-intersecting parasagittal arcs. Target localization begins with identification and contouring of target axial dimensions using diagnostic CT scans. A system for processing these target contours has been produced using the Beam's Eye View tech-

nique. Projected target cross sectional areas at each gantry/table position are found by graphical search and each leaf of the collimator is set to position. The localization results are then sent to dosimetry calculation for the production of isodose plots on the three principal planes, individual field cross plots, dose volume histograms, and complication probability functions. The three dimensional dosimetry technique developed here is termed the modified negative field method.

Dosimetry from cases previously treated using the University of Florida Stereotactic Radiosurgery System are compared with the developed method and also with a rotating collimator system that has been described in the literature. It is shown that the method developed herein provides better conformation and homogeneity in dose throughout the target volume than those techniques used at present.

DTIC QUALITY INSPECTED 8

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	

## REFERENCES

- Arc85 G. Arcovito, A. Piermattei, G. D'Abramo, F.A. Bassi, "Dose measurements and calculations of small radiation fields for 9-MV x rays," Med. Phys. 12:779 (1985)
- Bet84 O.O. Betti, V.E. Derechinsky, "Hyperselective encephalic irradiation with linear accelerator," Acta Neurochirurgica Suppl. 33:385 (1984)
- Bja82 B.E. Bjarngard, L.H. Brown, G.K. Svensson, "Scatter dose decrement values for rectangular fields," Med. Phys. 9:830 (1982)
- Bja90 B.E. Bjarngard, J.S. Tsai, R.K. Rice, "Doses on the central axes of narrow 6-MV x-ray beams," Med. Phys. 17:794 (1990)
- Boe90 R. Boesecke, T. Bruckner, G. Ende, "Landmark based correlation of medical images," Phys. Med. Biol. 35:121 (1990)
- Bor90 T. Bortfeld, J. Burkelbach, R. Boesecke, W. Schlegel, "Methods of image reconstruction from projections applied to conformation radiotherapy," Phys. Med. Biol. 35:1423 (1990)
- Bov90 F.J. Bova, "Radiation Physics," Neurosurgery Clinics of North America 1:909 (1990)
- Bov91 F.J. Bova, W.J. Friedman, "Stereotactic angiography: An inadequate database for radiosurgery?" Int. J. Radiation Oncology Biol. Phys. 20:891 (1991)
- Boy85 A. Boyer, E. Mok, "A photon dose distribution model employing convolution calculations," Med. Phys. 12:169 (1985)
- Boy86 A.L. Boyer, E.C. Mok, "Calculation of photon dose distributions in an inhomogeneous medium using convolutions," Med. Phys. 13:503 (1986)



- Chi81 L.M. Chin, P.K. Kijewski, G.K. Svensson, J.T. Chaffey, M.B. Levene, B.E. Bjarngard, "A computer-controlled radiation therapy machine for pelvic and para-aortic nodal areas," *Int. J. Radiation Oncology Biol. Phys.* 7:61 (1981)
- Chu86 C.S. Chui, R. Mohan, "Off-center ratios for three-dimensional dose calculations," *Med. Phys.* 13:409 (1986)
- Chu88a C.S. Chui, R. Mohan, D. Fontanella, "Dose computations for asymmetric fields defined by independent jaws," *Med. Phys.* 15:92 (1988)
- Chu88b C.S. Chui, R. Mohan, "Extraction of pencil beam kernels by the deconvolution method," *Med. Phys.* 15:138 (1988)
- Fit75 L.T. Fitzgerald, W. Mauderli, "Analysis of errors in three-dimensional reconstruction of radium implants from stereo radiographs," *Radiology* 115:455 (1975)
- Fli89 J.C. Flickinger, "An integrated logistic formula for prediction of complications from radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 12:879 (1989)
- Fli90a J.C. Flickinger, L.D. Lunsford, A. Wu, A.H. Maitz, A.M. Kalend, "Treatment planning for gamma knife radiosurgery with multiple isocenters," *Int. J. Radiation Oncology Biol. Phys.* 18:1495 (1990)
- Fli90b J.C. Flickinger, M.C. Schell, D.A. Larson, "Estimation of complications for linear accelerator radiosurgery with the integrated logistic formula," *Int. J. Radiation Oncology Biol. Phys.* 19:143 (1990)
- Fli90c J.C. Flickinger, A. Maitz, A. Kalend, L.D. Lunsford, A. Wu, "Treatment volume shaping with selective beam blocking using the Leksell gamma unit," *Int. J. Radiation Oncology Biol. Phys.* 19:783 (1990)
- Fol82 J.D. Foley, A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading MA (1984)
- Fri89a W.A. Friedman, F.J. Bova, "Stereotactic radiosurgery," *Contemporary Neurosurgery* 11:1 (1989)
- Fri89b W.A. Friedman, F.J. Bova, "The University of Florida radiosurgery system," *Surg. Neurol.* 32:334 (1989)

- Fri90 W.A. Friedman, "LINAC radiosurgery," *Neurosurgery Clinics of North America* 1:991 (1990)
- Gil90 P.L. Gildenberg, "The history of stereotactic neurosurgery," *Neurosurgery Clinics of North America* 1:765 (1990)
- Har85 G.H. Hartmann, W. Schlegel, V. Sturm, B. Kober, O. Pastyr, W.J. Lorenz, "Cerebral radiation surgery using moving field irradiation at a linear accelerator facility," *Int. J. Radiation Oncology Biol. Phys.* 11:1185 (1985)
- Iwa85 A. Iwasaki, "A method of calculating high-energy photon primary absorbed dose in water using forward and backward spread dose-distribution functions," *Med. Phys.* 12:731 (1985)
- Ker80 J.G. Kereiakes, M. Rosenstein, CRC Handbook of Radiation Doses in Nuclear Medicine and Diagnostic X-Ray, CRC Press, Boca Raton FL (1980)
- Kha70 F.M. Khan, "Computer dosimetry of partially blocked fields in cobalt teletherapy," *Radiology* 97:405 (1970)
- Kha80 F.M. Khan, W. Sewchand, J. Lee, J.F. Williamson, "Revision of tissue-maximum ratio and scatter-maximum ratio concepts for cobalt 60 and higher energy x-ray beams," *Med. Phys.* 7:230 (1980)
- Kha84 F.M. Khan, The Physics of Radiation Therapy, Williams and Wilkins, Baltimore MD (1984)
- Lar90 D.A. Larson, P.H. Gutin, "Introduction to radiosurgery," *Neurosurgery Clinics of North America* 1:897 (1990)
- Lea87 D.D. Leavitt, J.R. Stewart, "Optimization of electron arc therapy doses by dynamic collimator control," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lea89 D.D. Leavitt, J.R. Stewart, J.H. Moeller, W.L. Lee, G.A. Takach Jr., "Electron arc therapy: Design, implementation and evaluation of a dynamic multi-vane collimator system," *Int. J. Radiation Oncology Biol. Phys.* 17:1089 (1989)

- Lea91 D.D. Leavitt, F.A. Gibbs Jr., M.P. Heilbrun, J.H. Moeller, G.A. Takach Jr., "Dynamic field shaping to optimize stereotactic radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 21:1247 (1991)
- Lek51 L. Leksell, "The stereotaxic method and radiosurgery of the brain," *Acta Chir. Scand.* 102:316 (1951)
- Low90 N.N. Low, S. Vijayakumar, I. Rosenberg, S. Rubin, R. Virudachalam, D.R. Spelbring, G.T.Y. Chen, "Beam's eye view based prostate treatment planning: Is it useful?" *Int. J. Radiation Oncology Biol. Phys.* 19:759 (1990)
- Lul87 B. Lulu, "Conversion of CT coordinates to stereotaxic frame coordinates for the Brown-Roberts-Wells frame," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lut88 W. Lutz, K.R. Winston, N. Maleki, "A system for stereotactic radiosurgery with a linear accelerator," *Int. J. Radiation Oncology Biol. Phys.* 14:373 (1988)
- Man77 J. Mantel, H. Perry, "Automatic variation of field size and dose rate in rotation therapy," *Int. J. Radiation Oncology Biol. Phys.* 2:697 (1977)
- Met89 C.E. Metz, L.E. Fencil, "Determination of three-dimensional structure in biplane radiography without prior knowledge of the relationship between the two views: Theory," *Med. Phys.* 16:45 (1989)
- Moh87 R. Mohan, C.S. Chui, "Use of fast Fourier transforms in calculating dose distributions for irregularly shaped fields for three dimensional treatment planning," *Med. Phys.* 14:70 (1987)
- Moh88 R. Mohan, G. Barest, L.J. Brewster, C.S. Chui, G.J. Kutcher, J.S. Laughlin, Z. Fuks, "A comprehensive three-dimensional radiation treatment planning system," *Int. J. Radiation Oncology Biol. Phys.* 15:481 (1988)
- Myr88 L.C. Myriantopoulos, G.T.Y. Chen, D.R. Spelbring, C.A. Pelizzari, I. Rosenberg, S. Vijayakumar, H.J. Halpern, M. Reese, "Beam's eye view volumetrics: An aid in rapid treatment plan optimization," 30th Annual Meeting of the American Society for Therapeutic Radiology and Oncology, Manuscript (1988)

- Phi91 Philips Medical Systems, Stereotactic Radiosurgery Operations Manual, Release 1.12.12, Philips Medical Systems U.K. (1991)
- Pik87 B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, "Dose distributions in dynamic stereotactic radiosurgery," *Med. Phys.* 14:780 (1987)
- Pik90 G.B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, A. Olivier, L. Souhami, "Dose distributions in radiosurgery," *Med. Phys.* 7:296 (1990)
- Pre88 W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge U.K. (1988)
- Ric87a R.K. Rice, J.L. Hansen, G.K. Svensson, R.L. Siddon, "Measurement of dose distributions in small beams of 6 MV x-rays," *Phys. Med. Biol.* 32:1087 (1987)
- Ric87b J. Richter, P. Klemm, M. Neumann, G. Nowak, O. Sauer, "Computer control of a linac for dynamic treatment," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Saw87a C.B. Saw, K. Ayyangar, N. Suntharalingam, "Coordinate transformations and calculation of the angular and depth parameters for a stereotactic system," *Med. Phys.* 14:1042 (1987)
- Saw87b C.B. Saw, K. Ayyangar, N. Suntharalingam, "A simple algorithm for calculating the angular coordinates and depth for the BRW stereotaxic system," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sid86 R.L. Siddon, "3-D treatment planning algorithms," 1986 American Association of Physicists in Medicine Summer School Proceedings, 533 (1986)
- Sid87 R.L. Siddon, N.H. Barth, "Stereotaxic localization of intracranial targets," *Int. J. Radiation Oncology Biol. Phys.* 13:1241 (1987)
- Sou90 L. Souhami, A. Olivier, E.B. Podgorsak, M. Pla, G.B. Pike, "Radiosurgery of cerebral arteriovenous malformations with the dynamic stereotactic irradiation," *Int. J. Radiation Oncology Biol. Phys.* 19:775 (1990)

- Spe87 D.R. Spelbring, G.T.Y. Chen, C.A. Pelizarri, A.Awan, H. Sutton, "A computer simulated evaluation of multi-vane collimator beam delivery systems," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sta88 G. Starkschall, "A convolution method for constructing primary beam profiles in the presence of beam modifiers," Med. Phys. 15:657 (1988)
- Suh90 T.S. Suh, "Optimization of dose distribution for the system of linear accelerator based stereotactic radiosurgery," Ph.D. Dissertation, University of Florida (1990)
- Toe90 K.D. Toennies, J.K. Udupa, G.T. Herman, I.L. Wornam, S.R. Buchman, "Registration of 3D objects and surfaces," IEEE Computer Graphics and Applications 10:52 (1990)
- Win88 K.R. Winston, W. Lutz, "Linear accelerator as a neurosurgical tool for stereotactic radiosurgery," Neurosurgery 22:454 (1988)

Abstract of Dissertation Presented to the Graduate  
School of the University of Florida in Partial Fulfillment  
of the Requirements for the Degree of Doctor of Philosophy

CONFORMAL STEREOTACTIC RADIOSURGERY  
WITH MULTILEAF COLLIMATION

by

Dale C. Moss  
Captain, USAF, BSC

1992

352 pages

This paper outlines and implements a method of producing dose distributions that conform to any arbitrary, irregularly shaped target by means of dynamic conformal collimation using a multileaf collimator. The method may be summarized in three steps: production of the treatment plan; localization of the target; and the calculation of three dimensional dosimetry. Provision is made for volumetric evaluation of dosimetry with dose volume histograms and complication probability functions.

Treatment plans follow the standard of non-conformal stereotactic radiosurgery with several non-intersecting parasagittal arcs. Target localization begins with identification and contouring of target axial dimensions using diagnostic CT scans. A system for processing these target contours has been produced using the Beam's Eye View tech-

nique. Projected target cross sectional areas at each gantry/table position are found by graphical search and each leaf of the collimator is set to position. The localization results are then sent to dosimetry calculation for the production of isodose plots on the three principal planes, individual field cross plots, dose volume histograms, and complication probability functions. The three dimensional dosimetry technique developed here is termed the modified negative field method.

Dosimetry from cases previously treated using the University of Florida Stereotactic Radiosurgery System are compared with the developed method and also with a rotating collimator system that has been described in the literature. It is shown that the method developed herein provides better conformation and homogeneity in dose throughout the target volume than those techniques used at present.

## REFERENCES

- Arc85 G. Arcovito, A. Piermattei, G. D'Abramo, F.A. Bassi, "Dose measurements and calculations of small radiation fields for 9-MV x rays," Med. Phys. 12:779 (1985)
- Bet84 O.O. Betti, V.E. Derechinsky, "Hyperselective encephalic irradiation with linear accelerator," Acta Neurochirurgica Suppl. 33:385 (1984)
- Bja82 B.E. Bjarngard, L.H. Brown, G.K. Svensson, "Scatter dose decrement values for rectangular fields," Med. Phys. 9:830 (1982)
- Bja90 B.E. Bjarngard, J.S. Tsai, R.K. Rice, "Doses on the central axes of narrow 6-MV x-ray beams," Med. Phys. 17:794 (1990)
- Boe90 R. Boesecke, T. Bruckner, G. Ende, "Landmark based correlation of medical images," Phys. Med. Biol. 35:121 (1990)
- Bor90 T. Bortfeld, J. Burkelbach, R. Boesecke, W. Schlegel, "Methods of image reconstruction from projections applied to conformation radiotherapy," Phys. Med. Biol. 35:1423 (1990)
- Bov90 F.J. Bova, "Radiation Physics," Neurosurgery Clinics of North America 1:909 (1990)
- Bov91 F.J. Bova, W.J. Friedman, "Stereotactic angiography: An inadequate database for radiosurgery?" Int. J. Radiation Oncology Biol. Phys. 20:891 (1991)
- Boy85 A. Boyer, E. Mok, "A photon dose distribution model employing convolution calculations," Med. Phys. 12:169 (1985)
- Boy86 A.L. Boyer, E.C. Mok, "Calculation of photon dose distributions in an inhomogeneous medium using convolutions," Med. Phys. 13:503 (1986)



- Chi81 L.M. Chin, P.K. Kijewski, G.K. Svensson, J.T. Chaffey, M.B. Levene, B.E. Bjarngard, "A computer-controlled radiation therapy machine for pelvic and para-aortic nodal areas," *Int. J. Radiation Oncology Biol. Phys.* 7:61 (1981)
- Chu86 C.S. Chui, R. Mohan, "Off-center ratios for three-dimensional dose calculations," *Med. Phys.* 13:409 (1986)
- Chu88a C.S. Chui, R. Mohan, D. Fontanella, "Dose computations for asymmetric fields defined by independent jaws," *Med. Phys.* 15:92 (1988)
- Chu88b C.S. Chui, R. Mohan, "Extraction of pencil beam kernels by the deconvolution method," *Med. Phys.* 15:138 (1988)
- Fit75 L.T. Fitzgerald, W. Mauderli, "Analysis of errors in three-dimensional reconstruction of radium implants from stereo radiographs," *Radiology* 115:455 (1975)
- Fli89 J.C. Flickinger, "An integrated logistic formula for prediction of complications from radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 12:879 (1989)
- Fli90a J.C. Flickinger, L.D. Lunsford, A. Wu, A.H. Maitz, A.M. Kalend, "Treatment planning for gamma knife radiosurgery with multiple isocenters," *Int. J. Radiation Oncology Biol. Phys.* 18:1495 (1990)
- Fli90b J.C. Flickinger, M.C. Schell, D.A. Larson, "Estimation of complications for linear accelerator radiosurgery with the integrated logistic formula," *Int. J. Radiation Oncology Biol. Phys.* 19:143 (1990)
- Fli90c J.C. Flickinger, A. Maitz, A. Kalend, L.D. Lunsford, A. Wu, "Treatment volume shaping with selective beam blocking using the Leksell gamma unit," *Int. J. Radiation Oncology Biol. Phys.* 19:783 (1990)
- Fol82 J.D. Foley, A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading MA (1984)
- Fri89a W.A. Friedman, F.J. Bova, "Stereotactic radiosurgery," *Contemporary Neurosurgery* 11:1 (1989)
- Fri89b W.A. Friedman, F.J. Bova, "The University of Florida radiosurgery system," *Surg. Neurol.* 32:334 (1989)

- Fri90 W.A. Friedman, "LINAC radiosurgery," *Neurosurgery Clinics of North America* 1:991 (1990)
- Gil90 P.L. Gildenberg, "The history of stereotactic neurosurgery," *Neurosurgery Clinics of North America* 1:765 (1990)
- Har85 G.H. Hartmann, W. Schlegel, V. Sturm, B. Kober, O. Pastyr, W.J. Lorenz, "Cerebral radiation surgery using moving field irradiation at a linear accelerator facility," *Int. J. Radiation Oncology Biol. Phys.* 11:1185 (1985)
- Iwa85 A. Iwasaki, "A method of calculating high-energy photon primary absorbed dose in water using forward and backward spread dose-distribution functions," *Med. Phys.* 12:731 (1985)
- Ker80 J.G. Kereiakes, M. Rosenstein, CRC Handbook of Radiation Doses in Nuclear Medicine and Diagnostic X-Ray, CRC Press, Boca Raton FL (1980)
- Kha70 F.M. Khan, "Computer dosimetry of partially blocked fields in cobalt teletherapy," *Radiology* 97:405 (1970)
- Kha80 F.M. Khan, W. Sewchand, J. Lee, J.F. Williamson, "Revision of tissue-maximum ratio and scatter-maximum ratio concepts for cobalt 60 and higher energy x-ray beams," *Med. Phys.* 7:230 (1980)
- Kha84 F.M. Khan, The Physics of Radiation Therapy, Williams and Wilkins, Baltimore MD (1984)
- Lar90 D.A. Larson, P.H. Gutin, "Introduction to radiosurgery," *Neurosurgery Clinics of North America* 1:897 (1990)
- Lea87 D.D. Leavitt, J.R. Stewart, "Optimization of electron arc therapy doses by dynamic collimator control," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lea89 D.D. Leavitt, J.R. Stewart, J.H. Moeller, W.L. Lee, G.A. Takach Jr., "Electron arc therapy: Design, implementation and evaluation of a dynamic multi-vane collimator system," *Int. J. Radiation Oncology Biol. Phys.* 17:1089 (1989)

- Lea91 D.D. Leavitt, F.A. Gibbs Jr., M.P. Heilbrun, J.H. Moeller, G.A. Takach Jr., "Dynamic field shaping to optimize stereotactic radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 21:1247 (1991)
- Lek51 L. Leksell, "The stereotaxic method and radiosurgery of the brain," *Acta Chir. Scand.* 102:316 (1951)
- Low90 N.N. Low, S. Vijayakumar, I. Rosenberg, S. Rubin, R. Virudachalam, D.R. Spelbring, G.T.Y. Chen, "Beam's eye view based prostate treatment planning: Is it useful?" *Int. J. Radiation Oncology Biol. Phys.* 19:759 (1990)
- Lul87 B. Lulu, "Conversion of CT coordinates to stereotaxic frame coordinates for the Brown-Roberts-Wells frame," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lut88 W. Lutz, K.R. Winston, N. Maleki, "A system for stereotactic radiosurgery with a linear accelerator," *Int. J. Radiation Oncology Biol. Phys.* 14:373 (1988)
- Man77 J. Mantel, H. Perry, "Automatic variation of field size and dose rate in rotation therapy," *Int. J. Radiation Oncology Biol. Phys.* 2:697 (1977)
- Met89 C.E. Metz, L.E. Fencil, "Determination of three-dimensional structure in biplane radiography without prior knowledge of the relationship between the two views: Theory," *Med. Phys.* 16:45 (1989)
- Moh87 R. Mohan, C.S. Chui, "Use of fast Fourier transforms in calculating dose distributions for irregularly shaped fields for three dimensional treatment planning," *Med. Phys.* 14:70 (1987)
- Moh88 R. Mohan, G. Barest, L.J. Brewster, C.S. Chui, G.J. Kutcher, J.S. Laughlin, Z. Fuks, "A comprehensive three-dimensional radiation treatment planning system," *Int. J. Radiation Oncology Biol. Phys.* 15:481 (1988)
- Myr88 L.C. Myriantopoulos, G.T.Y. Chen, D.R. Spelbring, C.A. Pelizzari, I. Rosenberg, S. Vijayakumar, H.J. Halpern, M. Reese, "Beam's eye view volumetrics: An aid in rapid treatment plan optimization," 30th Annual Meeting of the American Society for Therapeutic Radiology and Oncology, Manuscript (1988)

- Phi91 Philips Medical Systems, Stereotactic Radiosurgery Operations Manual, Release 1.12.12, Philips Medical Systems U.K. (1991)
- Pik87 B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, "Dose distributions in dynamic stereotactic radiosurgery," Med. Phys. 14:780 (1987)
- Pik90 G.B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, A. Olivier, L. Souhami, "Dose distributions in radio-surgery," Med. Phys. 7:296 (1990)
- Pre88 W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge U.K. (1988)
- Ric87a R.K. Rice, J.L. Hansen, G.K. Svensson, R.L. Siddon, "Measurement of dose distributions in small beams of 6 MV x-rays," Phys. Med. Biol. 32:1087 (1987)
- Ric87b J. Richter, P. Klemm, M. Neumann, G. Nowak, O. Sauer, "Computer control of a linac for dynamic treatment," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Saw87a C.B. Saw, K. Ayyangar, N. Suntharalingam, "Coordinate transformations and calculation of the angular and depth parameters for a stereotactic system," Med. Phys. 14:1042 (1987)
- Saw87b C.B. Saw, K. Ayyangar, N. Suntharalingam, "A simple algorithm for calculating the angular coordinates and depth for the BRW stereotaxic system," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sid86 R.L. Siddon, "3-D treatment planning algorithms," 1986 American Association of Physicists in Medicine Summer School Proceedings, 533 (1986)
- Sid87 R.L. Siddon, N.H. Barth, "Stereotaxic localization of intracranial targets," Int. J. Radiation Oncology Biol. Phys. 13:1241 (1987)
- Sou90 L. Souhami, A. Olivier, E.B. Podgorsak, M. Pla, G.B. Pike, "Radiosurgery of cerebral arteriovenous malformations with the dynamic stereotactic irradiation," Int. J. Radiation Oncology Biol. Phys. 19:775 (1990)

- Spe87 D.R. Spelbring, G.T.Y. Chen, C.A. Pelizarri, A.Awan, H. Sutton, "A computer simulated evaluation of multi-vane collimator beam delivery systems," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sta88 G. Starkschall, "A convolution method for constructing primary beam profiles in the presence of beam modifiers," Med. Phys. 15:657 (1988)
- Suh90 T.S. Suh, "Optimization of dose distribution for the system of linear accelerator based stereotactic radiosurgery," Ph.D. Dissertation, University of Florida (1990)
- Toe90 K.D. Toennies, J.K. Udupa, G.T. Herman, I.L. Wornam, S.R. Buchman, "Registration of 3D objects and surfaces," IEEE Computer Graphics and Applications 10:52 (1990)
- Win88 K.R. Winston, W. Lutz, "Linear accelerator as a neurosurgical tool for stereotactic radiosurgery," Neurosurgery 22:454 (1988)

CONFORMAL STEREOTACTIC RADIOSURGERY  
WITH MULTILEAF COLLIMATION

BY

DALE C. MOSS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1992

## TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT . . . . .	iv
CHAPTERS	
1 INTRODUCTION . . . . .	1
Stereotactic Radiosurgery . . . . .	1
Field Shaping . . . . .	3
Dynamic Conformal Collimation . . . . .	4
2 REVIEW OF THE LITERATURE . . . . .	6
Stereotactic Radiosurgery Systems . . . . .	7
Conformal Collimation . . . . .	9
Target Localization . . . . .	12
Photon Dosimetry . . . . .	14
Verification . . . . .	16
3 THE STEREOTACTIC PROCEDURE AT THE UNIVERSITY OF FLORIDA . . . . .	18
Equipment . . . . .	18
Patient Preparation . . . . .	20
Target Localization . . . . .	20
Treatment Planning . . . . .	21
Patient Treatment . . . . .	24
4 TARGET DEFINITION . . . . .	25
The Rotation Operation . . . . .	25
Stereotactic and Beam's Eye View Coordinates . . . . .	29
Coordinate Transformation and Target Rotation . . . . .	31
Target Localization . . . . .	46
5 INVESTIGATIONAL DOSIMETRY . . . . .	73
The Convolution Method . . . . .	74
The Negative Field Method . . . . .	79
Conclusion . . . . .	84

6	THE MODIFIED NEGATIVE FIELD METHOD . . . . .	108
	Output Factor . . . . .	110
	Tissue Maximum Ratio . . . . .	111
	Primary Off Center Ratio . . . . .	113
	Boundary Factor . . . . .	117
	The Dosimetry Calculation Process . . . . .	120
	Dosimetry Results . . . . .	124
7	DOSIMETRY VERIFICATION . . . . .	133
	Irregular Field Block Construction . . . . .	133
	Computerized Film Dosimetry . . . . .	135
	Irregular Small Field Dose Model Verification . . . . .	137
	Analysis Results . . . . .	141
8	COLLIMATOR SPECIFICATION . . . . .	155
	Leaf Shape . . . . .	155
	Leaf Width . . . . .	156
	Arc Compression . . . . .	158
	Gantry Incrementation . . . . .	158
	Localization Margin . . . . .	159
	Conclusion . . . . .	160
9	CLINICAL EXAMPLES . . . . .	226
	Case 1 . . . . .	227
	Case 2 . . . . .	228
	Case 3 . . . . .	229
	Case 4 . . . . .	230
	Conclusion . . . . .	232
10	CONCLUSION . . . . .	258
APPENDICES		
A	TREATMENT PLANNING . . . . .	261
B	TARGET LOCALIZATION PROGRAM . . . . .	269
C	DOSE MODEL PROGRAM (PLANE) . . . . .	296
D	DOSE MODEL PROGRAM (VOLUME) . . . . .	319
E	INTEGRATED LOGISTIC FUNCTION PROGRAM . . . . .	337
REFERENCES . . . . .		345
BIOGRAPHICAL SKETCH . . . . .		351



Abstract of Dissertation Presented to the Graduate  
School of the University of Florida in Partial Fulfillment  
of the Requirements for the Degree of Doctor of Philosophy

CONFORMAL STEREOTACTIC RADIOSURGERY  
WITH MULTILEAF COLLIMATION

by

Dale C. Moss

August, 1992

Chairman: Frank J. Bova, Ph.D.

Major Department: Nuclear Engineering Sciences

This paper outlines and implements a method of producing dose distributions that conform to any arbitrary, irregularly shaped target by means of dynamic conformal collimation using a multileaf collimator. The method may be summarized in three steps: production of the treatment plan; localization of the target; and the calculation of three dimensional dosimetry. Provision is made for volumetric evaluation of dosimetry with dose volume histograms and complication probability functions.

Treatment plans follow the standard of non-conformal stereotactic radiosurgery with several non-intersecting parasagittal arcs. A  $2\pi$  geometry with at least five arcs of  $100^\circ$  each has been determined to be optimal.

Target localization begins with identification and contouring of target axial dimensions using diagnostic CT

scans. A system for processing these target contours has been produced using the Beam's Eye View technique. Projected target cross sectional areas at each gantry/table position are found, after appropriate translation/rotation, by graphical search and each leaf of the collimator is set to position. Optimal localization parameters have been determined.

The localization results are then sent to dosimetry calculation for the production of isodose plots on the three principal planes, individual field cross plots, dose volume histograms, and complication probability functions. The three dimensional dosimetry technique developed here is termed the modified negative field method, as individual fields with the dimension of each leaf are subtracted from an open field defined by the combined dimensions of all of the leaves.

Dosimetry from cases previously treated using the University of Florida Stereotactic Radiosurgery System are compared with the developed method and also with a rotating collimator system that has been described in the literature. It is shown that the method developed herein provides better conformation and homogeneity in dose throughout the target volume than those techniques used at present.

## CHAPTER 1 INTRODUCTION

### Stereotactic Radiosurgery

At its inception, stereotactic radiosurgery was considered to be a tool to remove a lesion from the treatment area by delivering a radiation dose sufficient to cause tissue necrosis. Hartmann observed: "If the dose gradient is very steep . . . a single dose that is sufficient to cause necrosis of the tissue volume selected can be administered." [Har85, pg. 1185]. For this dose, ". . . the most important factor . . . is the physically determined concentration of the radiation on the target, in contrast to fractionated radiotherapy, which is based on differences in radiosensitivity between tumor cells and cells of the adjacent healthy tissue." [Har85, pg. 1185]. Further, ". . . the side effects of a single high dose irradiation can only be avoided or minimized if the area of tissue chosen for irradiation is precisely anatomically defined and adjusted . . ." [Har85, pg. 1186].

More recently, particularly in the case of vascular lesions, it has been acknowledged that tissue necrosis does not take place in the irradiated tissue (although some tumors may become necrotic after treatment). After treat-

ment of a vascular lesion, through an incompletely understood process, the nidus of the lesion thromboses, after which the lesion disappears over a period of approximately two years. Satisfactory results appear best achieved by producing a homogeneous dose distribution throughout the lesion [Lea91]. To this end, a successful stereotactic radiosurgery system must incorporate hardware and software to determine the size and location of the lesion, to plan the treatment, and to deliver the radiation in accordance with the plan [Win88].

Radiosurgery is used on patients who are not good candidates for conventional neurosurgery for whatever reason. Many radiosurgery treatments have been for the obliteration of arteriovenous malformations (Steiner reports that 85% of such lesions treated are undetectable by angiography two years later [Win88]). Other lesions reported treated include acoustic neurinomas, anaplastic astrocytomas, low grade astrocytomas, carotid-cavernous fistulas, cavernous angiomas, choroid plexus papillomas, craniopharyngiomas, ependymomas, germinomas, glioblastomas, lymphomas, medulloblastomas, meningiomas, metastases, oligodendrogliomas, pineoblastomas, pineocytomas, pituitary tumors, and venous angiomas [Lar90].

The auxiliary collimator used in linear accelerator based stereotactic radiosurgery is seen by most as an important component, as it both precisely shapes the radiation

beam, difficult with the main collimators alone, and reduces beam penumbra by being physically closer to the target. The circular auxiliary collimators generally used produce a spherical dose distribution at isocenter. Rectangular collimators produce a cylindrical dose distribution [Suh90] which may be of use in certain cases. In either case, the dose distribution produced may or may not correspond to the shape of the target.

#### Field Shaping

The aim of optimizing dose distributions by conforming the distributions to the shape of the target is summarized in four requirements for conventional large-field teletherapy [Bor90] which may be equally well applied to stereotactic radiosurgery:

- a. The dose applied to the target should be very close to the prescribed dose.
- b. The dose should be homogeneously distributed across the target.
- c. The dose to the organs at risk should be less than the tolerated maximum dose.
- d. In the tissue surrounding the target, the dose should be low.

Various standard and nonstandard schemes of combining open fields exist in practice to conform the dose to the target. In standard large-field teletherapy, parallel opposed open fields are weighted to move the combined field

isodose distribution in the direction of the more heavily weighted field. Angled opposed fields are used to form a triangular distribution with the apex bisecting the angle between the beams (with equal beam weighting) or any arbitrary resecting angle (with unequal weighting).

Custom collimators have been employed effectively in large field teletherapy as additions to the main (secondary) system collimators. These beam blocks are manufactured to the physician's specification after exploratory x-rays of the potential treatment area are analyzed. The physician draws the outline of the target on the x-ray and blocks (usually of Lipowitz' metal) are made to match the outline, taking into account any magnification and divergence.

#### Dynamic Conformal Collimation

In stereotactic radiosurgery, the normal open circular beams may also be weighted to shift the distribution. The standard arc orientations may be shifted to the same end. Attempts at better conformal collimation may be undertaken by using multiple isocenters and/or differing collimator sizes in different arcs [Suh90]. This latter method is effective and is used in practice, but is costly in both planning and treatment time and complexity. It also results in hot spots in the lesion in volumes where the dose distributions from each isocenter overlap.

Multileaf collimators are used to shape the beam profile to the target projected area either statically or

dynamically. Static multileaf collimation simply replaces custom made beam blocks in normal teletherapy. Dynamic multileaf collimation is much more challenging. In this mode, the collimator is adjusted to conform to the projected area of the target as the patient is moved on the table and/or the gantry is rotated around the patient.

Leavitt et al. [Lea91, pg. 1249] cite Nedze et al. having ". . . shown that tumor dose inhomogeneity introduced through the use of multiple isocenters is the strongest variable predicting complication, and have emphasized the need for development of alternative treatment techniques for lesions otherwise requiring multiple isocenter techniques." The goal of this work then is to develop a stereotactic radiosurgical technique to shape the dose distribution to the target volume in such a manner that the dose distribution throughout that volume is homogeneous, the target volume is enclosed in the prescribed isodose volume, and the dose drops off rapidly outside of the defined treatment volume. In support of this goal, research has been conducted in target localization, multileaf collimator vane specification, three-dimensional irregular field photon dosimetry (both plane dosimetry and dose volume histogram), and guidelines for optimization through treatment plan variation. The end product should be useful clinically.

## CHAPTER 2 REVIEW OF THE LITERATURE

To recapitulate the elements necessary for a functioning radiosurgery system we recall the strictures set forth in Chapter 1 [Bor90]:

- a. The dose applied to the target should be very close to the prescribed dose.
- b. The dose should be homogeneously distributed across the target.
- c. The dose to the organs at risk should be less than the tolerated maximum dose.
- d. In the tissue surrounding the target, the dose should be low.

With these guidelines in mind, the literature was reviewed to gain insights in five categories in support of this research. Stereotactic radiosurgery systems that have been described were assimilated for historical background and differing technique. Conformal collimation methods in both large and small field radiotherapy were investigated, the large field methods being generally applicable, while the small field methods were specifically oriented to stereotactic radiosurgery. Target localization procedures were assessed as a necessary adjunct to conformal collimation



since computed rotational dosimetry uses superposition of discrete beam locations to simulate the integration of a moving radiation beam, and therefore the target must be able to be visualized at any possible gantry/table position combination. Historical and contemporary photon dosimetry algorithms were analyzed as an important aspect, as the concept of "virtual target" is fully realized in this treatment modality. Finally, verification strategies were looked at as support for any dosimetry system adopted.

#### Stereotactic Radiosurgery Systems

Stereotactic radiosurgery was first suggested by Leksell in 1951 [Lek51]. He initially used 200 to 300 kVp x-rays, then charged particles. He finally settled on collimated  $^{60}\text{Co}$  beams, first 179 and later 201, the number used in the present Leksell Gamma Knife™. This device was conceived as a non-invasive method for performing functional neurosurgery, a course which was later abandoned for various considerations [Gil90]. Research then began on the treatment of intracranial lesions. The Gamma Knife is in clinical use today in many centers.

Other teams began experimenting with charged particle beams in the late 1950's at sites in Sweden, Berkeley, and Boston (studies which continue to the present). These beams of proton or helium ions are produced by synchrocyclotrons, and take advantage of the Bragg peak of the particles to focus the beams at depth [Pik87].

While these methods are undeniably effective, they also require the use of very expensive, dedicated equipment found only in a few large medical and research centers. Betti and his coworkers appear to be the first to develop a linear accelerator as a treatment machine, reported in 1984, used in Buenos Aires since 1982, with circular brass auxiliary collimators [Bet84]. Colombo and associates reported in 1985 on a linear accelerator based system in use in Vicenza, Italy, since 1982, with only the linear accelerator collimator jaws used to define the treatment area [Fri90].

The treatment accuracy of a linac radiosurgery system is dependent on the linac mechanical accuracy (how well the central axis of the rotating beam continuously coincides with the rotation isocenter), and on the target localization accuracy (how accurately the target is located with combinations of biplanar angiography, computed tomography, and magnetic resonance imaging). Hartmann, in Heidelberg, reported on his use of a linear accelerator as a treatment machine in 1985 [Har85]. He used an auxiliary collimator coupled directly to the head of the linear accelerator, with the consequence that gantry sag limited the accuracy of the delivered dose (the treatment accuracy) to approximately  $\pm 2.0$  mm. Lutz et al. [Lut88] also used this type of auxiliary collimator, with variously sized inserts, and reported a treatment accuracy of  $\pm 2.4$  mm in any direction at the 95% confidence level. Souhami et al. [Sou90] used simultaneous

rotation of the linac gantry and couch with lead auxiliary collimators without reporting accuracy.

Friedman and Bova [Fri89b] developed a three-axis sliding bearing system to couple the auxiliary collimator to the linear accelerator head, thus avoiding the effects of gantry sag and improving the dose delivery of the University of Florida Stereotactic Radiosurgery System to an average mechanical accuracy of 0.2 mm. To date, this is the most precise and accurate system of those that have been described in the literature, and is commercially available as the Philips SRS 200 Stereotactic Radiosurgery System.

#### Conformal Collimation

A simple translational four-leaf collimator system was described by Chin [Chi81]. In his system, the beam is swept superior to inferior over the treatment volume by motion of the table under the beam. The two-leaf set parallel to the axial plane is narrowed to a slit, and the perpendicular set of leaves are adjusted to coincide with the projected target edges. This produces distributions which conform well to the target and can be specifically set to avoid sensitive structures.

Spelbring et al. [Spe87] performed a computer simulation of large field teletherapy multivane collimator systems that showed an advantage for these systems on a case specific basis. Leavitt et al. also investigated the use of dynamic multivane collimators in electron arc therapy com-

putationally for the purposes of optimization [Lea87], and by hardware realization [Lea89]. Both computation and measurement found conformal collimation to be effective in improving homogeneity in target dose while restricting high dose areas to the target, computationally by an average of 11% for areas covered by the 100% dose line to 15% for areas covered by the 90% line, confirmed by measurements using the prototype multivane collimator.

Flickinger et al. [Fli90c] have studied conformal collimation as applied to the Leksell Gamma Knife. By blocking various patterns of the 201 <sup>60</sup>Co beams dose volume shapes may be altered from the normal spherical distributions. Examples of calculational dosimetry are given. Left and right lateral ports may be plugged to produce distributions that are extended in the axial direction (cephalad/caudad). An AP strip in the sagittal plane may be plugged to produce distributions that are extended laterally. A lateral strip in the coronal plane may be plugged to extend distributions anterior to posterior. All these are used to shape distributions to extended ellipsoid targets. An example of film dosimetry shows good fit to calculated high isodose lines.

It has been suggested that rectangular collimators, employed with rotation to follow major target axes, be used to improve distributions in stereotactic radiosurgery [Suh90]. A rectangular target was followed with the Beam's

Eye View technique to define the direction of the target's major axis and projected target dimensions at each gantry/table position. A standard four arc plan (equally spaced arcs) was generated using a cylindrical dose model. This plan was compared, using a dose volume histogram, to a plan employing two isocenters to cover the same target volume. The histogram showed a slight improvement of dosimetry with the Beam's Eye View technique, although dose homogeneity was not addressed.

A simple four blade rotating conformal collimator was constructed and described by Leavitt et al. [Lea91]. The collimator was a double layer design with two leaves in each layer. The layers and the leaves in the layers could be rotated to best fit the target projected area. The Beam's Eye View technique was used to follow the target through all gantry/table angles and to automatically adjust the leaf positions at each increment of angle. An irregular field dosimetry system was developed and was evaluated against measured distributions with both film and diode. Isodose plots were then compared between the standard circular collimator, the conformal vane collimator, and the dual isocenter techniques. This showed a 24% improvement in the amount of normal brain covered by the 80% isodose line in favor of the conformal technique when compared to the circular, and a 1% improvement in the same volume in favor of the conformal technique when compared to the dual isocenter. It

was noted that concave shapes could not be effectively fit using this technique.

There is no multileaf collimator system described in the literature for the case of small field linac based stereotactic radiosurgery. The system, here proposed, would be capable of fitting concave or other irregularly shaped targets limited only by the size of the leaf and by the treatment margin desired.

#### Target Localization

Prior to a stereotactic treatment of any kind it is necessary to precisely locate the target. This is done with angiography, computerized tomography, and magnetic resonance imaging, alone or in any combination [Fri89a]. At present, localization consists of obtaining two orthogonal views in planes which best describe the target and using these views to determine the target isocenter.

Stereotactic localization of targets has been the object of many presentations. Siddon and Barth [Sid87] reported on the use of the BRW frame to obtain isocenter and orthogonal view data. Their method reports the ability to localize a target pointer to within 0.3 mm using a modified localizer box. In two papers, Saw et al. [Saw87a; Saw87b] gave a system of calculations using a standard BRW frame for the purpose of stereotactic neurosurgical implants, although accuracy of placement is not shown. Lulu [Lul87] published a description of a system, also using the standard BRW

localizer, for transforming CT coordinates to BRW coordinates for localization. This system uses basic geometrical transformations made possible by the unique positioning information imparted by axial slices of the localizer. Localization errors are reported to be between 0.5 and 1 CT pixel width.

Visualization of the target after localization is necessary to observe dosimetry presentations, usually in the form of isodose lines. The target and dosimetry information must be viewed from any angle in three-dimensions to ensure coverage of the target and sparing of critical organs that may be in close proximity. Fitzgerald and Mauderli [Fit75] analyzed the errors in three-dimensional reconstruction of implant dosimetry using stereo-radiography. Metz and Fencil [Met89] developed a method of showing three-dimensional structure based on two different but arbitrarily oriented radiographs. Boesecke et al. [Boe90] and Toennies et al. [Toe90] used prominent bony landmarks to register and visualize their targets when rotated.

The Beam's Eye View (BEV) technique is useful in target visualization under dynamic conditions. This technique is based on the acquisition of target data such that the target may be viewed along the radiation path through the collimator as the gantry and table rotate about the target, located at the rotation isocenter of the system. Mohan et al. [Moh88] have incorporated the BEV technique as an integral

part of a complete three-dimensional treatment planning system. Myrianthopoulos et al. [Myr88] and Low et al. [Low90] both presented BEV rotational methods coupled with volume analysis to determine adequacy of target coverage in dynamic radiotherapy.

Dynamic localization is necessary for dynamic conformal collimation. There is no method described in the literature for localizing a rotating target such that its projected area may be defined by a multileaf collimator.

#### Photon Dosimetry

Photon beam dose models are many and widespread. Most of these models incorporate primary dose (from primary photons), secondary dose (from scattered photons), off-axis ratios (for points off of the central axis of the incident beam), percent depth dose or tissue maximum ratio (to account for exponential falloff of the beam intensity in tissue), and relative output factors (to correct for field sizes other than that calibrated).

Small field dose models incorporate these factors to a greater or lesser extent. Bjarngard et al. [Bja82] derived an analytical term for the scatter component of the small beam, which was averaged over the radius of the beam. Chui et al. [Chu86] use off-axis ratios derived from a product of backscatter factors. Hartmann et al. [Har85] subsume all these factors into measured dose profiles and depth dose curves. Pike et al. [Pik87] rely on percent depth doses,



off-axis ratios, and inverse square corrections and shows measured and calculated dose distributions [Pik90]. Bova [Bov90] uses TMR and OAR tables in the University of Florida stereotactic radiosurgery system.

Bjarngard et al. [Bja90] observe that for small 6 MV x-ray beams the central axis dose is significantly reduced for fields of less than 1 cm radius due to electron disequilibrium, that photons scattered from the collimator do not affect dose, and that only very small beams of less than 0.07 cm radius are affected by source size induced penumbra. Khan et al. [Kha80] allude to the idea that scatter dose is of little effect in small beams, while Arcovito et al. [Arc85] and Rice et al. [Ric87] specifically allow for and calculate a scatter correction factor for small 9 and 6 MV x-ray beams, respectively.

Perhaps the most interesting are the convolution models. Boyer and Mok [Boy85] and Iwasaki [Iwa85] use these models to provide a fast method of completely describing an incident photon beam energy distribution. Boyer and Mok [Boy86] extended their method to calculate distributions in inhomogeneous media. Mohan et al. [Moh87] and Starkschall [Sta88] use convolutions of pencil beam profiles with irregular field shapes by Fourier transform operations to arrive at dose distributions. These models use nothing but the primary dose modified by simple factors derived from the convolution operations.

### Verification

Calibrated ion chambers are the primary measurement tool in radiation therapy. After a beam is calibrated with an ion chamber, the secondary methods of film and diode dosimetry are used. The latter are secondary because they rely on pre-calibration with known beams to derive fitting factors that allow the calculation of unknown doses.

Ion chambers, as a standard, are accurate and precise, and can measure unknown field quantities without recourse to prior knowledge about the field. However, they are difficult to use with small fields, as they must be carefully aligned so that the full chamber volume is irradiated. Rice et al. [Ric87a] approached this problem by aligning the central axis of the chamber parallel to and coincident with the beam central axis, significantly reducing the required lateral coverage.

Films have advantages over both ion chambers and diodes in that they record a continua of data points versus a single point for chambers and diodes, and that their data collection is a permanent record that may be re-analyzed in light of new data as opposed to the "one-shot" nature of the other methods. Films are, however, sensitive to handling and processing variables. Bjarngard et al. [Bja90] have found that small field densitometry with film is a satisfactory tool.

Diodes are compact, reproduce well, and may be remotely read out in real time. They are sensitive to placement, however, and may give inaccurate readings if not oriented correctly in the radiation beam. They are also physically sensitive and prone to catastrophic failure.

Each of these methods has its place and each will be used to provide data for and to verify the dosimetry methods developed in this work.

CHAPTER 3  
THE STEREOTACTIC PROCEDURE AT THE UNIVERSITY OF FLORIDA

Equipment

A standard linear accelerator is used at the University of Florida for stereotactic radiosurgery. It is modified by the addition of a head stand for the Brown-Roberts-Wells™ (BRW) stereotactic ring (a conventional piece of neurosurgery apparatus), a shortened couch top to clear the head stand, and a bearing/holder system for auxiliary collimation (figure 3-1). Setup of the system modifications takes 10 to 15 minutes.

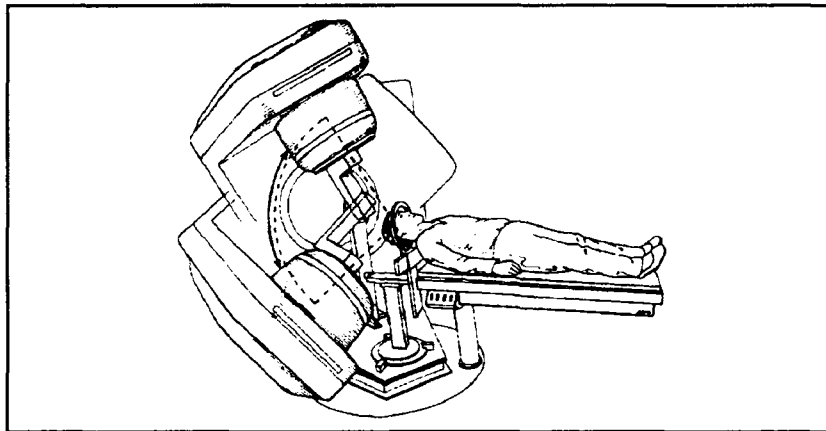


Figure 3-1: University of Florida Stereotactic Radiosurgery System ([Fri90, page 993], used with author's permission)

The BRW ring is the reference point for all localization of targets. It is a metal ring which is fixed to the

patient's head with stainless steel pins. The top surface of the ring is placed inferior to the target position and becomes the reference point for all localization and calculation.

The head stand and bearing/holder system are incorporated in a single portable piece of equipment that is positioned under the gantry of the linear accelerator such that the target can be placed accurately at the rotation isocenter of the radiation beam. The BRW ring is rigidly and precisely attached to the head stand. Micrometer adjustments allow the positioning of the localized target center to coincide with the system isocenter. A two-bearing system mechanically connects the head stand to the gantry. One set serves to rotate the BRW ring in the table plane, keeping the target centered while the table is rotated. The second set couples the collimator system to the head stand by a swing arm around the axis of gantry rotation, allowing accurate and precise beam positioning.

The swing arm end, directly under the linear accelerator head, is the mount for the auxiliary collimators. The purpose of auxiliary collimation is to both precisely define the beam and diminish penumbra effects. These collimators are 15 cm thick Lipowitz' metal (beam transmission approximately 2%) with circular apertures ranging from 0.5 to 3 cm in diameter. The apertures are tapered to match the beam divergence. The auxiliary collimator is loosely coupled to

the linear accelerator by a three-axis sliding bearing mounted on the head. This sliding bearing system divorces the auxiliary collimator from any gantry torque induced by sag or gantry bearing inexactness, thus improving the accuracy of the dose delivery.

#### Patient Preparation

The stereotactic radiosurgery treatment is conducted on an outpatient basis. The patient is initially seen in clinic where the BRW ring is fixed to the head. This is done under local anaesthesia (mixed lidocaine and markane injection). The ring is pinned to the skull with stainless steel pins at each of the four injection sites. The BRW ring used is a standard ring rebuilt to tighter tolerances to accommodate the demands of radiosurgery.

#### Target Localization

Targets are localized depending on their type. Vascular targets, such as arteriovenous malformations (AVM's), are localized by contrast angiography and by computerized tomography (CT). Other targets employ CT localization only.

In angiographic localization, the BRW ring is attached to a mount placed on the table end. An angiographic localizer is attached to the ring. The localizer consists of four lucite panels (anterior, posterior, left, and right) with radio-opaque fiducial marks (four in each, eight per AP/lateral projection) as defined reference points [Sid87]. Contrast is injected and fast biplane films are taken. The

neuroradiologist and neurosurgeon select the AP and lateral films that best define the nidus of the AVM for treatment planning.

The setup for CT localization is similar, though with a different localizer being attached to the BRW ring. The CT localizer is made up of three pairs of parallel rods oriented on the patient's major axis, with angled rods between them. Transverse CT slices show six fixed rod profiles with the angled rod profiles at varying positions between the fixed. The positions of the angled rod profiles relative to the fixed uniquely locates that slice in BRW space. As the spacing between the rods is known, any object circumscribed by the localizer cage can be localized accurately and precisely [Lul87; Saw87a]. Radio-opaque contrast is injected to define the target. All the CT data is transferred to 9 track tape for treatment planning.

#### Treatment Planning

At the University of Florida, treatment planning begins by transferring localization information to the planning system. If angiography has been performed to locate the target, the biplane films are placed on a digitizer and the neurosurgeon enters the position of each fiducial mark and traces the AP and lateral nidus boundaries. The system computes a geometrical center and a center of mass for each nidus projection, which should closely match if the nidus has been outlined correctly [Bov91]. The best matching

center pair is used as the center of the target. The CT tape is then mounted and the axial slice images are transferred into the system. Starting at the top slice, the position of each of the localizing rods is defined. The system automatically steps through the remaining slices, finds each corresponding rod position, and registers each slice. If CT is the only localizing modality used, the neurosurgeon traces the target boundary in each of the axial, coronal, and sagittal planes, then selects two of the planes in which the target centers best match, as in the angiography case, to define the target center.

Table 3-1: Standard nine arc treatment plan

Arc Number	Collimator Size	Table Angle	Gantry Start	Gantry Stop	Arc Weight
1	10 mm	10°	30°	130°	1
2	10 mm	30°	30°	130°	1
3	10 mm	50°	30°	130°	1
4	10 mm	70°	30°	130°	1
5	10 mm	350°	230°	330°	1
6	10 mm	330°	230°	330°	1
7	10 mm	310°	230°	330°	1
8	10 mm	290°	230°	330°	1
9	10 mm	270°	230°	330°	1

An initial treatment plan is entered, consisting of the number of arcs, collimator size for each arc, arc orientation (table angle), arc start and stop angles (gantry angles), and arc weighting. A standard nine arc treatment



plan is shown in table 3-1. Note that the table angles describe nine equally spaced parasagittal arc positions, all arcs are  $100^\circ$  (gantry start to gantry stop angle), and all arc weights and collimator sizes are equal.

Plan variables may be changed as necessary. Changing the collimator size will change the diameter of the isodose lines. Moving or deleting table angles will change the shape of the dose distribution. For example, deleting the lateral arcs ( $10^\circ$ ,  $30^\circ$ ,  $350^\circ$ , and  $330^\circ$  table angles) will result in an axial extension and lateral contraction of the dose distribution. Setting different weighting on different arcs can also shift the distribution.

Multiple isocenters may be specified for extended or irregularly shaped targets, with each isocenter set to cover a portion of the volume. Problems with this approach involve increased treatment plan complexity, increased treatment time, and often severe dose inhomogeneity within the treatment volume. This, however, is the only current operational approach to conformal stereotactic radiosurgery at the University of Florida.

Isodose distributions are then calculated and may be viewed on any arbitrary slice, as well as dose profiles across any defined line and dose volume histograms within the treatment volume. At present, plan optimization is by the visual, iterative method which can, and frequently does, entail lengthy planning sessions.

Patient Treatment

The stereotactic radiosurgery system accessories are attached to the linear accelerator and the isocenter position is set on the head stand. Independently, a phantom target is set up with an isocenter matching that set on the head stand [Win88]. The phantom target is attached to the head stand and x-ray images of the phantom target are taken at various standard gantry and table positions. If the images show the phantom target in the center of the collimated beam ( $\pm 0.2$  mm) on each exposure, the headstand settings are considered correct.

The patient is then brought into the treatment room and attached to the headstand. Treatment proceeds as defined by the treatment plan. At the conclusion of treatment, the BRW ring is removed and the patient is free to leave. Follow up consultation and angiography takes place at regular intervals.

## CHAPTER 4

### TARGET DEFINITION

A necessary preliminary to dose planning for conformal collimation is the setting of the leaves of the multileaf collimator to the margins of the projected target cross-section at each of the arc increments for all of the specified arcs. The Beam's Eye View (BEV) method, employed by Mantel et al. [Man77] for conventional rotation teletherapy, is used as the basis for visualizing and specifying the target boundaries. A graphical search is used for finding the boundaries after the target has been drawn on the computer screen. Each leaf is then set to the limit found for that leaf's sector of coverage. As background to the full explication of the method developed here, a discussion of graphical translation/rotation systems from a basic text [Fol82] follows.

#### The Rotation Operation

Rotation of any discrete point about the origin of a coordinate system is a mathematical process that is shown in figure 4-1. Point  $P(x,y)$  is rotated to point  $P(x',y')$ .  $P(x,y)$  can be specified by the  $x$  and  $y$  coordinates computed from the angle  $\alpha$  and the distance to the origin  $d$ :

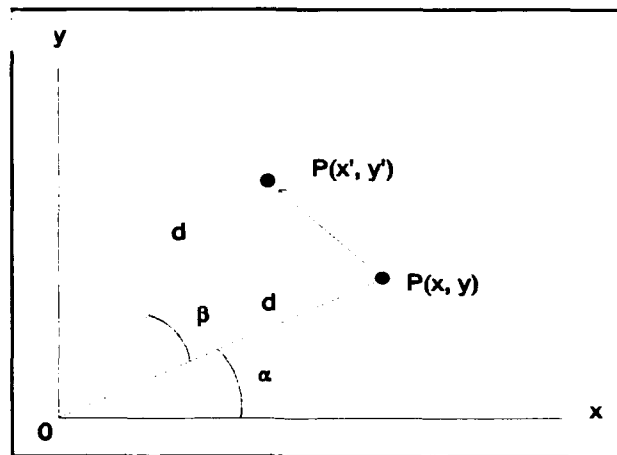


Figure 4-1: Point rotation about the origin

$$x = d \cos \alpha \quad (4-1)$$

$$y = d \sin \alpha \quad (4-2)$$

Point  $P(x', y')$  can then be seen to be simply:

$$x' = d \cos(\alpha + \beta) \quad (4-3)$$

$$y' = d \sin(\alpha + \beta) \quad (4-4)$$

Expanding by the sum-of-angles gives:

$$x' = d \cos \alpha \cos \beta - d \sin \alpha \sin \beta \quad (4-5)$$

$$y' = d \sin \alpha \cos \beta + d \cos \alpha \sin \beta \quad (4-6)$$

and by substituting the original formulas 4-1 and 4-2:

$$x' = x \cos \beta - y \sin \beta \quad (4-7)$$

$$y' = x \sin \beta + y \cos \beta \quad (4-8)$$

Adding the third dimension (the z axis positive perpendicular to and coming out of the page), we see that any rotation in x or y does not change the distance d from the z axis, therefore a rotation about the z axis simply results in all the points of rotation being multiplied by 1. This can be generalized to any rotation axis to result in the following sets of equations:

Rotation in the xy plane about the z axis:

$$x' = x \cos \theta - y \sin \theta \quad (4-9)$$

$$y' = x \sin \theta + y \cos \theta \quad (4-10)$$

$$z' = z \quad (4-11)$$

Rotation in the xz plane about the y axis:

$$x' = x \cos \theta + z \sin \theta \quad (4-12)$$

$$y' = y \quad (4-13)$$

$$z' = -x \sin \theta + z \cos \theta \quad (4-14)$$

Rotation in the yz plane about the x axis:

$$x' = x \quad (4-15)$$

$$y' = y \cos \theta - z \sin \theta \quad (4-16)$$

$$z' = y \sin \theta + z \cos \theta \quad (4-17)$$

It is the usual case to express these sets as matrix operations:

Rotation in the xy plane about the z axis:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z P \quad (4-18)$$

Rotation in the xz plane about the y axis:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_y P \quad (4-19)$$

Rotation in the yz plane about the x axis:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_x P \quad (4-20)$$

where  $R_{x/y/z}$  is the rotation operator about the x, y, or z axis and P is the orthogonalized representation of the point to be rotated (expressed in x, y, and z), respectively. By matrix multiplication, then, any combination of rotations about any combination of axes may be realized, recalling that matrix multiplications are not commutative (i.e.  $AB \neq BA$ ).

This suffices to rotate any point or group of points about the origin. To rotate about any arbitrary center, the rotation center must be first translated to the origin, the points rotated as previously described, and the origin translated back to the original rotation center. The translation is easily accomplished by subtracting the distance from the origin to the rotation center, properly orthogonalized, from all the points being translated (translation to the origin), and by adding the distance from the origin to the rotation center to all the points being translated (translation from the origin).

#### Stereotactic and Beam's Eye View Coordinates

Figure 4-2 shows the relationship between the stereotactic (BRW) and the Beam's Eye View (BEV) coordinate systems. The BRW coordinate system, with the axes axial (Ax, commonly called the vertical axis), lateral (Lat), and anterior/posterior (AP), has its origin centered in each of the three BRW localizer dimensions, and is fixed to and rotates with the table (rotation about the AP axis). This

system has the Cartesian coordinates  $xyz$  such that  $x$  is positive left lateral,  $y$  is positive anterior, and  $z$  is positive cranial [Lul87; Saw87a]. When fixed to the treatment table, the BRW location of the localized target is placed at rotation isocenter.

The BEV coordinate system has axes positive towards the gantry (GT), positive to the collimator left (AB), and positive towards the radiation source (UB, up beam). The origin is at rotation isocenter, is fixed to the collimator position, and rotates with the gantry (rotation about the GT axis). The BEV system is a generalization of that defined by Siddon [Sid86].

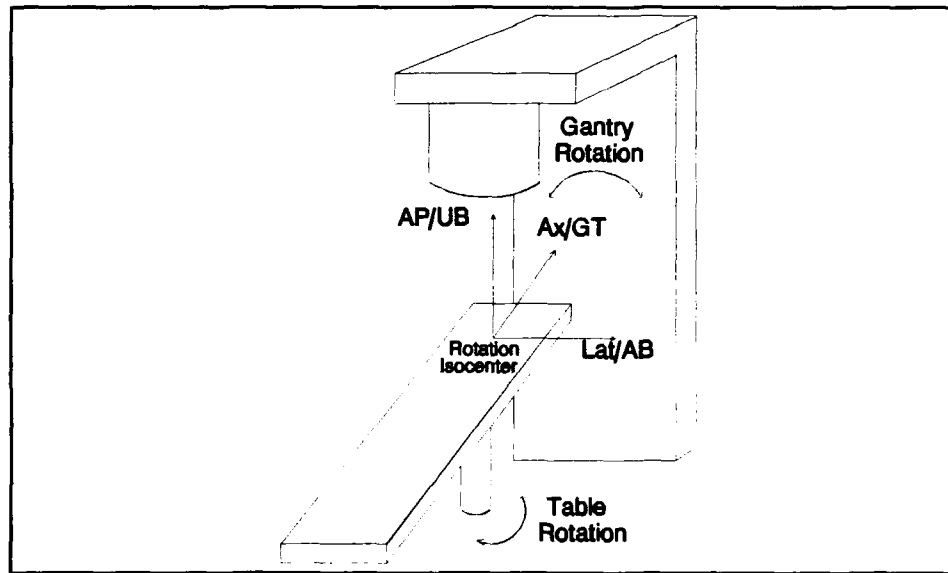


Figure 4-2: Stereotactic and beam's eye view coordinates

The target is captured in axial CT slices in the BRW coordinate system. This system must be transferred to the



BEV coordinate system to allow mapping of the target projected area as the table and gantry rotate about the designated target center.

#### Coordinate Transformation and Target Rotation

Recalling the non-commutative nature of matrix multiplications, care is necessary in the order of rotation, i.e. the order of axes about which the target is rotated, as the combined effect of rotating about several axes effectively results in a matrix multiplication process. In dealing with the many non-coplanar arcs of stereotactic radiosurgery as performed with the University of Florida system, two rotation axes are apparent; as the table rotates the target is rotated about the AP/UB axis, and as the gantry rotates the target is rotated about the Ax/GT axis, in this order.

One must be able to visualize rotation operations in three dimensions to arrive at this order of rotation. Consider the inverse, gantry rotation followed by table rotation. As the gantry rotates, in the BEV coordinate system the target counter rotates about the GT axis. If then followed by table rotation, still in the BEV coordinate system, the target must rotate about the UB and AB axes simultaneously, leading to complications in the mathematical treatment.

Consider, then, the stated order of rotation. As the table rotates, the target rotates about the UB axis. Then as the gantry rotates, the transformed target counter ro-

tates about the GT axis. Each of the operations is a previously defined rotation about a single axis. As a series of rotation operations results in a matrix multiplication, this combined operation may be expressed using equations 4-18 and 4-19 as  $R_{\text{table}} \cdot R_{\text{gantry}} \cdot P = R_{\text{SRS}} \cdot P$  with  $R_{\text{table}} \cdot R_{\text{gantry}}$  now defined as  $R_{\text{SRS}}$ , the SRS operator. Formally:

$$R_{\text{SRS}} = \begin{bmatrix} \cos\phi_t \cos\theta_g & -\sin\phi_t \cos\theta_g & \sin\theta_g \\ \sin\phi_t & \cos\phi_t & 0 \\ -\cos\phi_t \sin\theta_g & \sin\phi_t \sin\theta_g & \cos\theta_g \end{bmatrix} \quad (4-21)$$

where  $\phi_t$  is table rotation, and  $\theta_g$  is gantry rotation.

The SRS rotation process is then:

$$\begin{bmatrix} AB' \\ GT' \\ UB' \end{bmatrix} = R_{\text{SRS}} \begin{bmatrix} AB \\ GT \\ UB \end{bmatrix} \quad (4-22)$$

which may be orthogonalized as:

$$AB' = AB \cos\theta_t \cos\phi_g - GT \sin\theta_t \cos\phi_g + UB \sin\theta_g \quad (4-23)$$

$$GT' = AB \sin\phi_t + GT \cos\phi_t \quad (4-24)$$

$$UB' = -AB \cos\phi_t \sin\theta_g + GT \sin\phi_t \sin\theta_g + UB \cos\theta_g \quad (4-25)$$

Given a target positioned at rotation isocenter that is described in an axial series of CT slices, the operation of rotating the target to a series of BEV positions for localization follows this algorithm:

1) Map the BRW axes onto the BEV axes:

BEV AB axis = BRW Lateral axis

BEV GT axis = BRW Vertical axis

BEV UB axis = BRW AP axis

2) Convert data points defining the target in the BRW system and convert to BEV coordinates by:

BEV AB point = BRW Lateral point

BEV GT point = BRW Vertical point

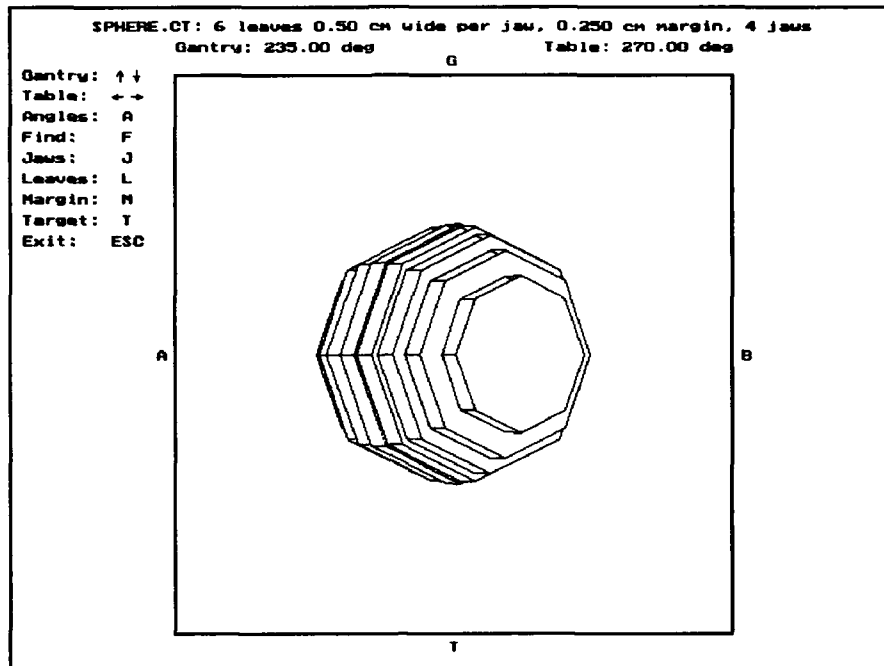
BEV UB point = BRW AP point

3) For each gantry position in each arc:

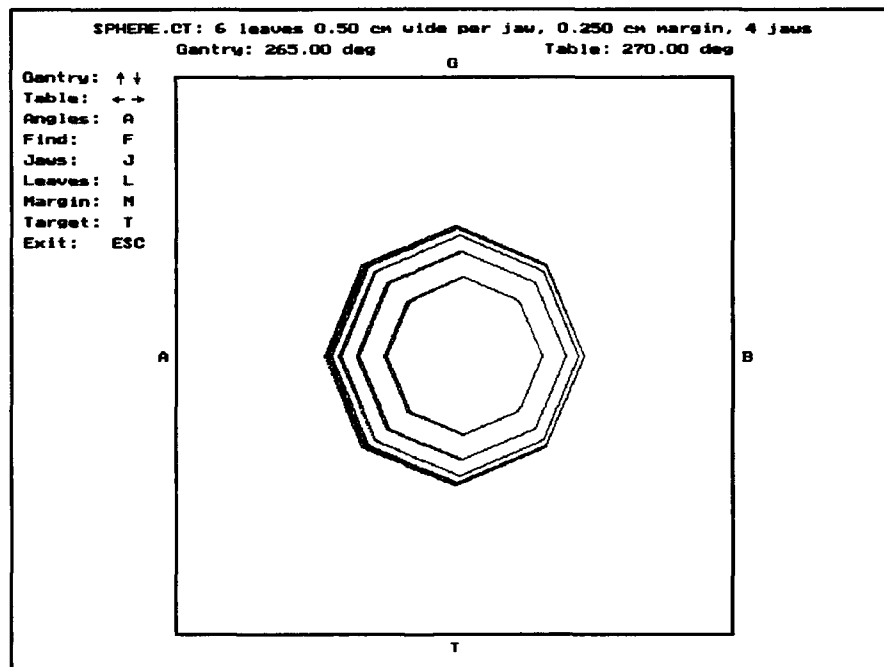
a) Translate the BRW isocenter to the rotation isocenter.

b) Rotate the target points with the SRS operator.

This process has been coded in the program LFLOC.C (appendix B) and is illustrated by figures 4-3 through 4-8 (from the program LFDEMO.C, a version of LFLOC.C, that removes hidden lines). The targets are, respectively, a sphere, an AP oriented ovoid, an axially oriented ovoid, a laterally oriented ovoid, an oblique ovoid, and a double oblique ovoid. All the targets are located at the center of the 20 cm diameter spherical head phantom. The sphere is 2 cm in diameter. The ovoids are 2 cm on the major axis, 1 cm on the minor axes. The AP, axial, and lateral ovoids have their major axis in the direction referenced. The oblique ovoid has its major axis in the sagittal plane, oriented 45° to the AP, resulting in an ovoid oriented from anterior

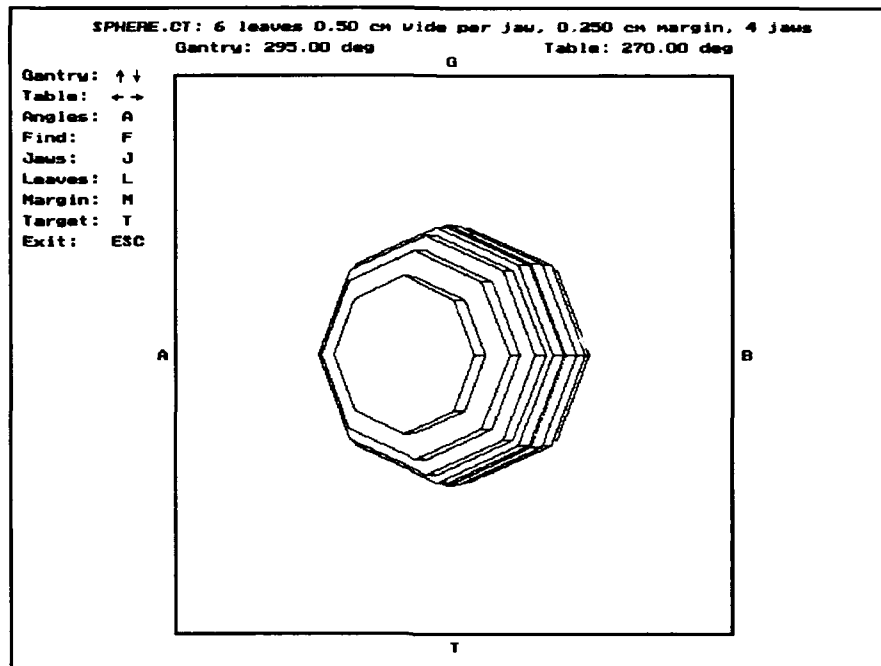


(a)

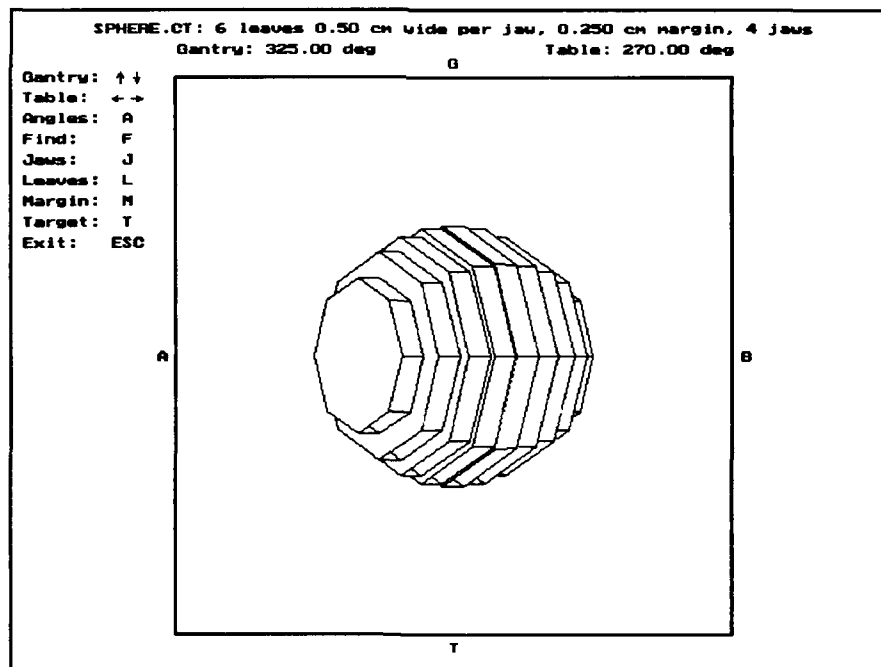


(b)

Figure 4-3: Sphere target rotation  
 (a) Gantry 235°; (b) Gantry 265°

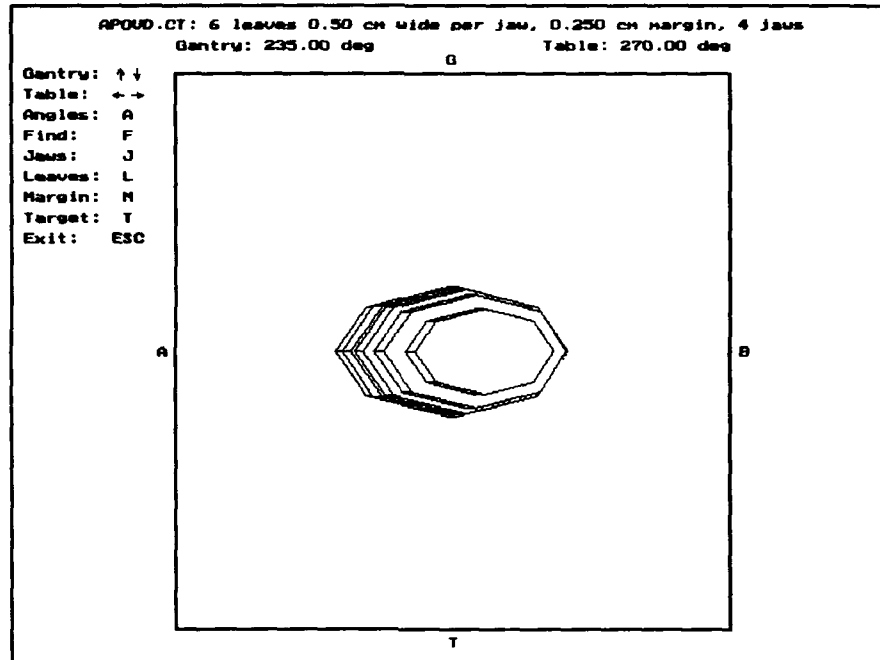


(c)

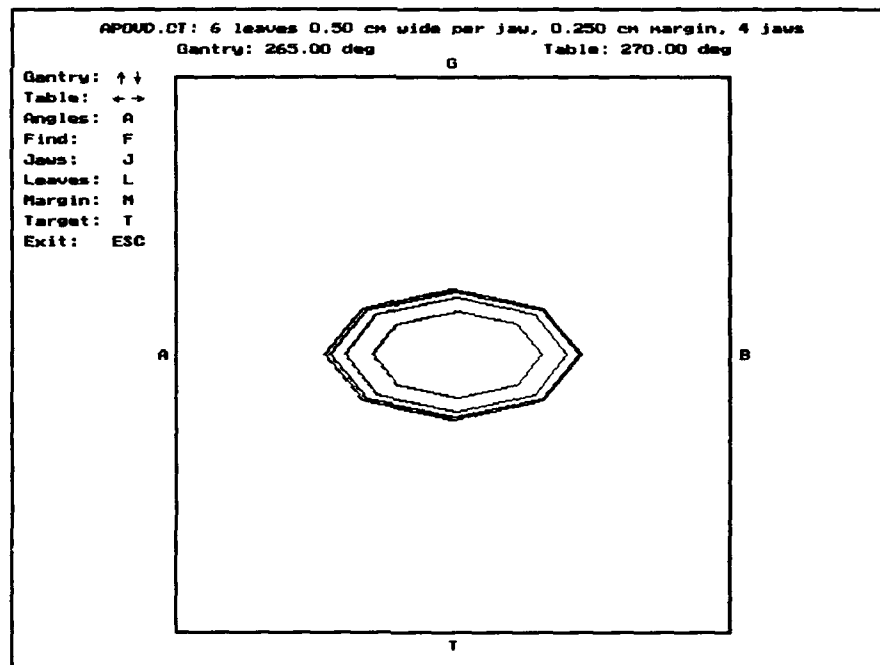


(d)

Figure 4-3 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

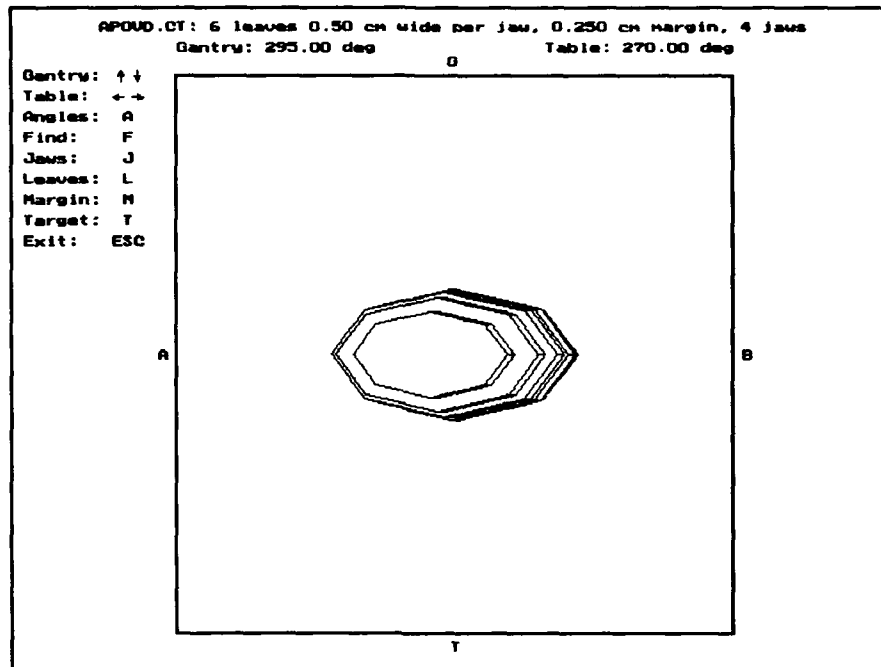


(a)

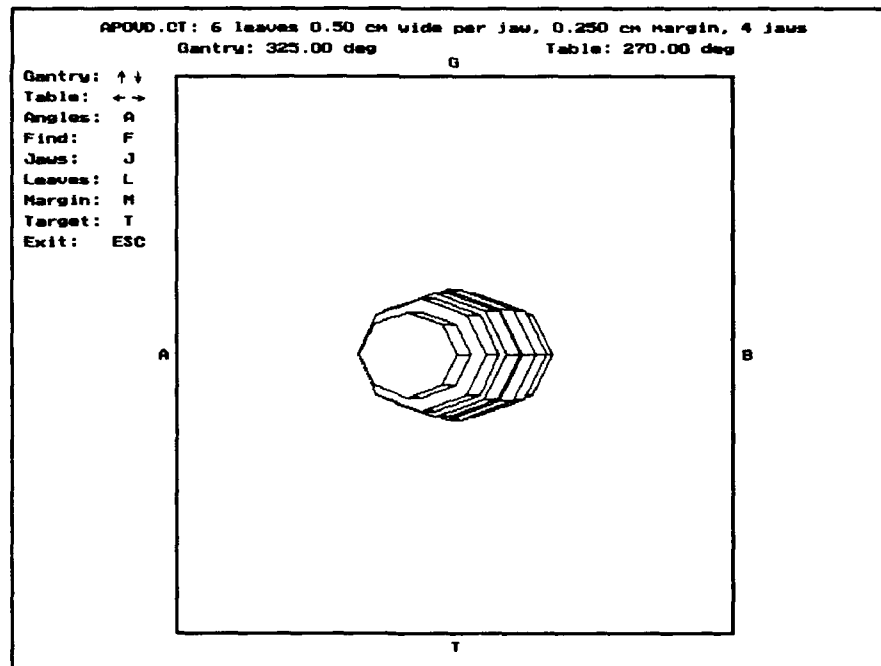


(b)

Figure 4-4: AP ovoid target rotation  
 (a) Gantry 235°; (b) Gantry 265°

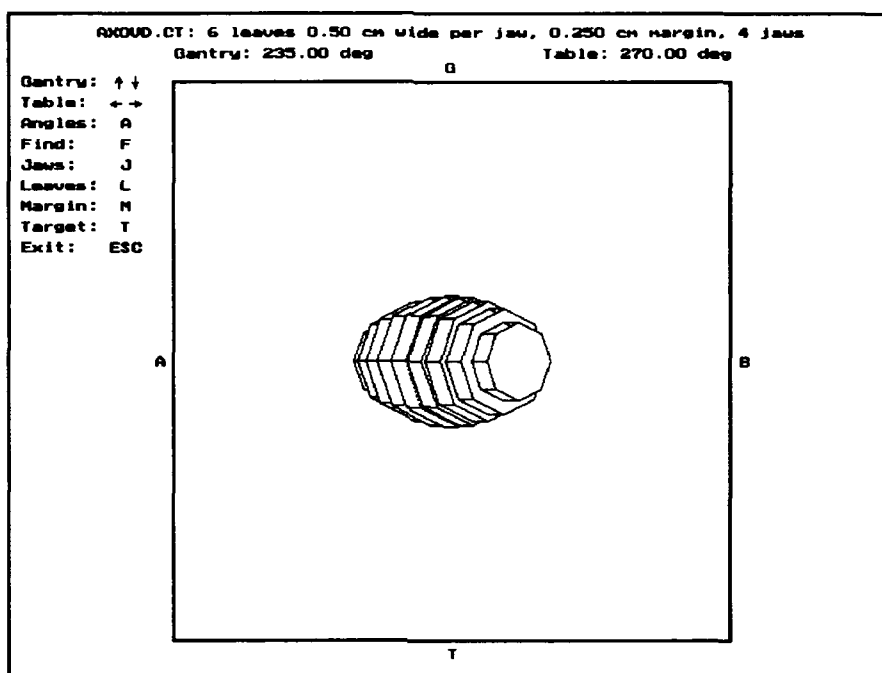


(c)

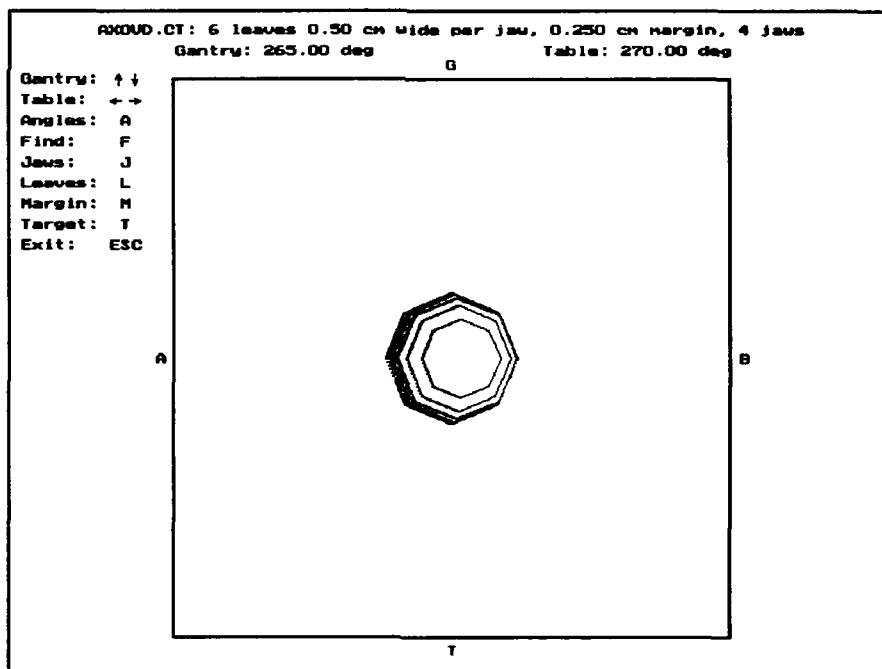


(d)

Figure 4-4 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



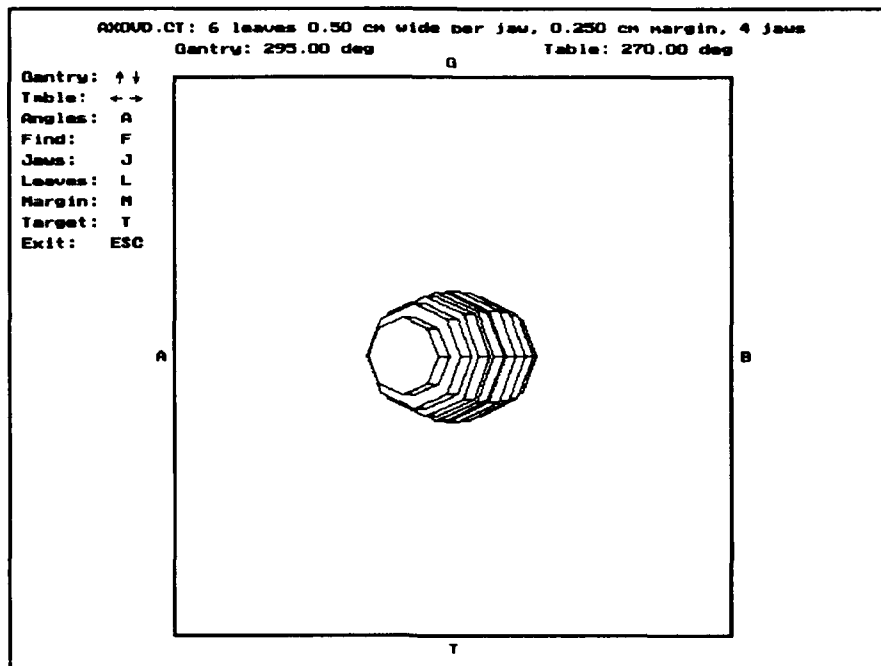
(a)



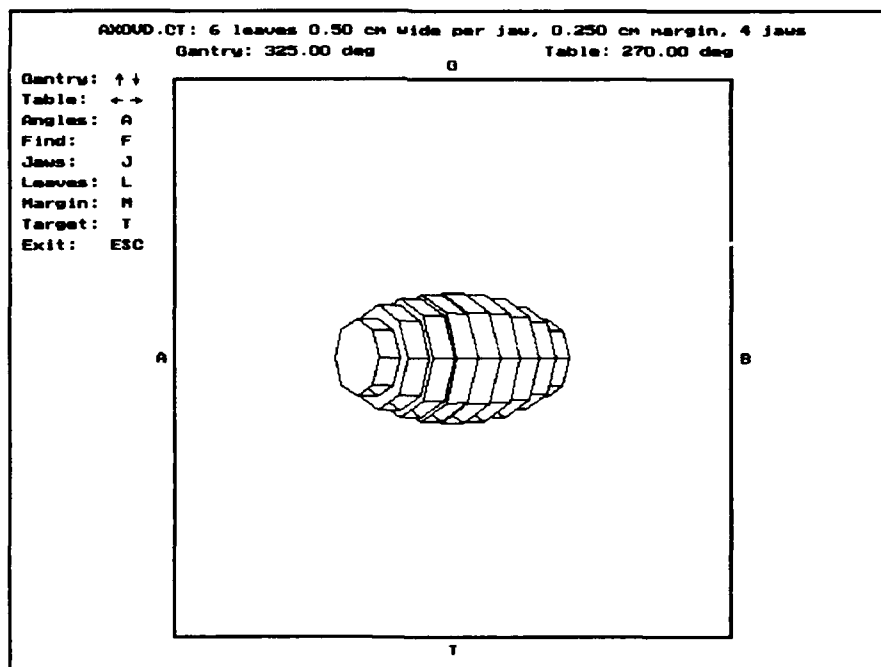
(b)

Figure 4-5: Axial ovoid target  
 (a) Gantry 235°; (b) Gantry 265°



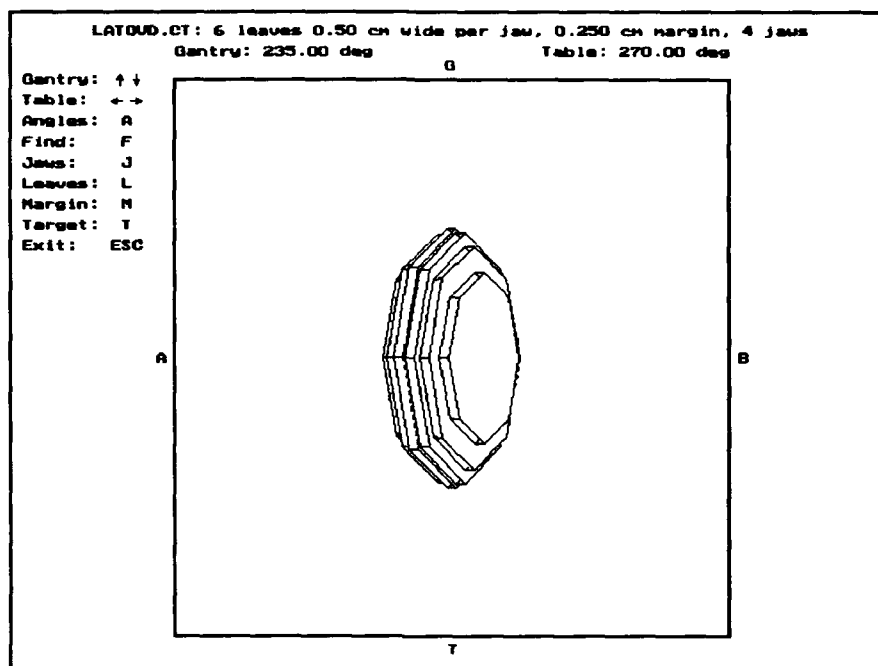


(c)

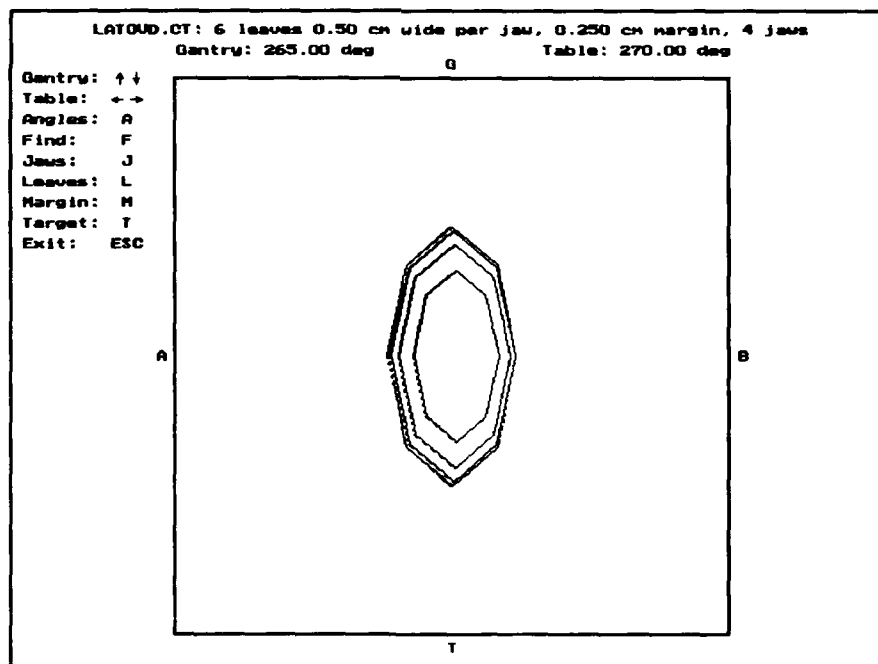


(d)

Figure 4-5 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

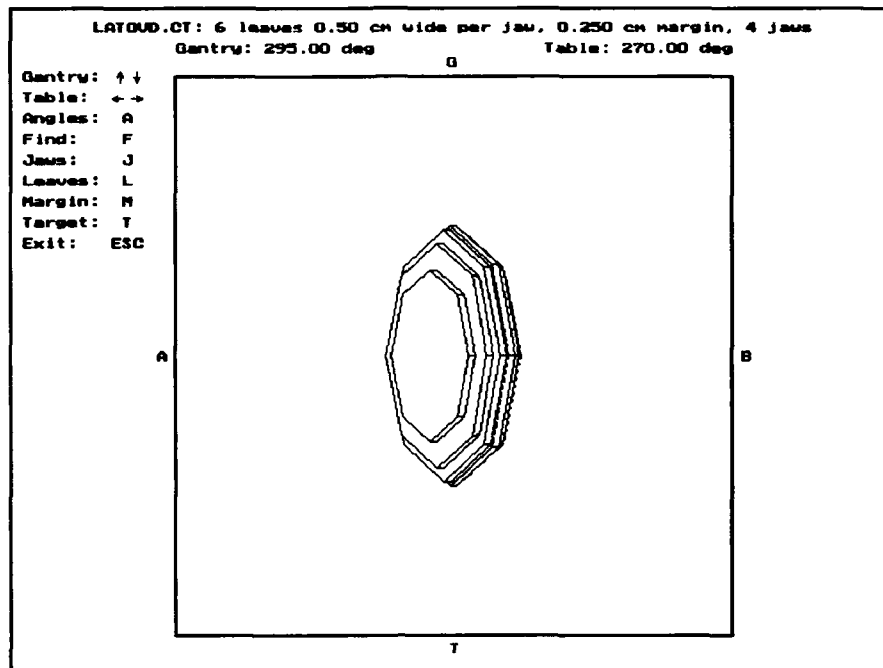


(a)

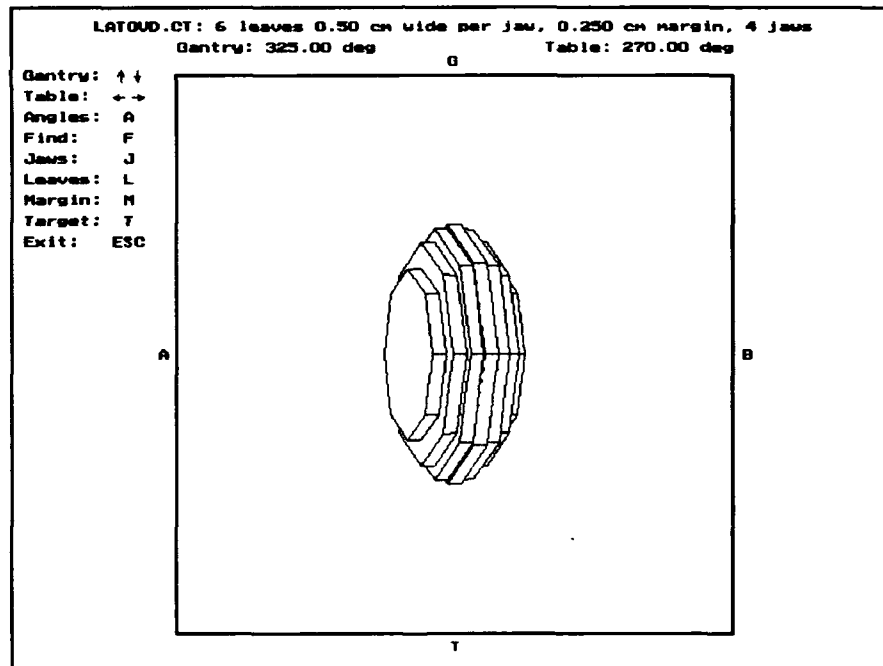


(b)

Figure 4-6: Lateral ovoid target  
 (a) Gantry 235°; (b) Gantry 265°

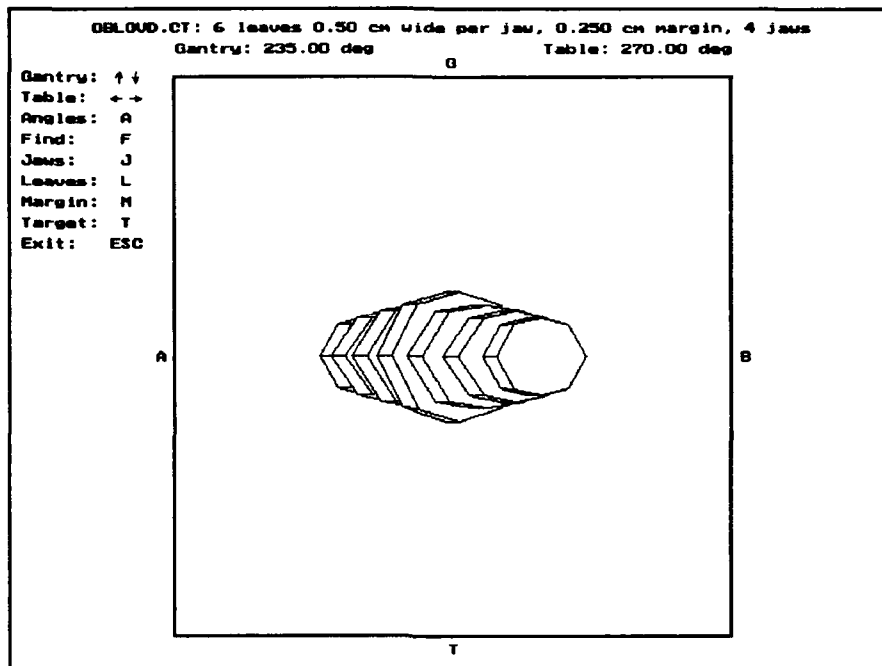


(c)

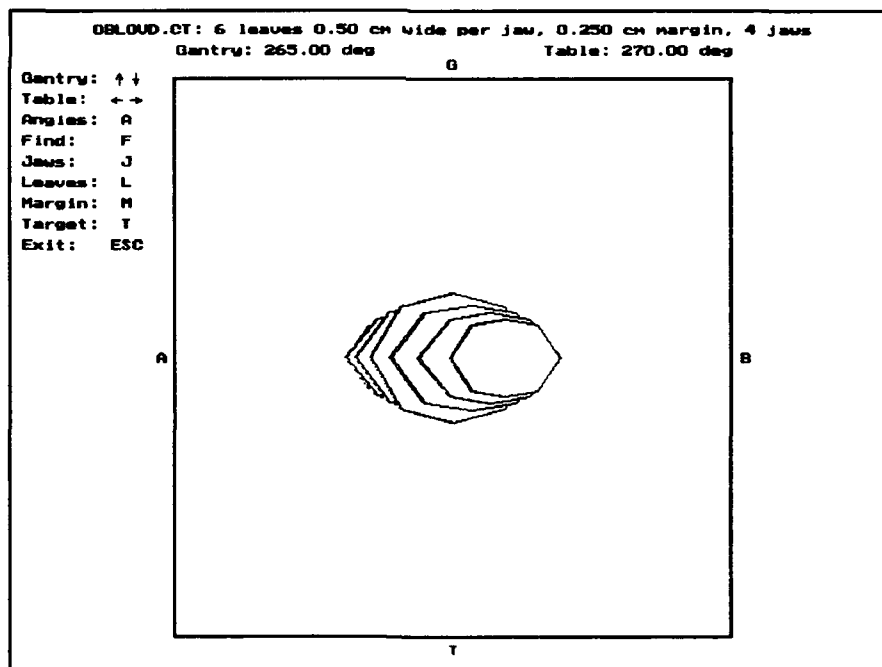


(d)

Figure 4-6 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

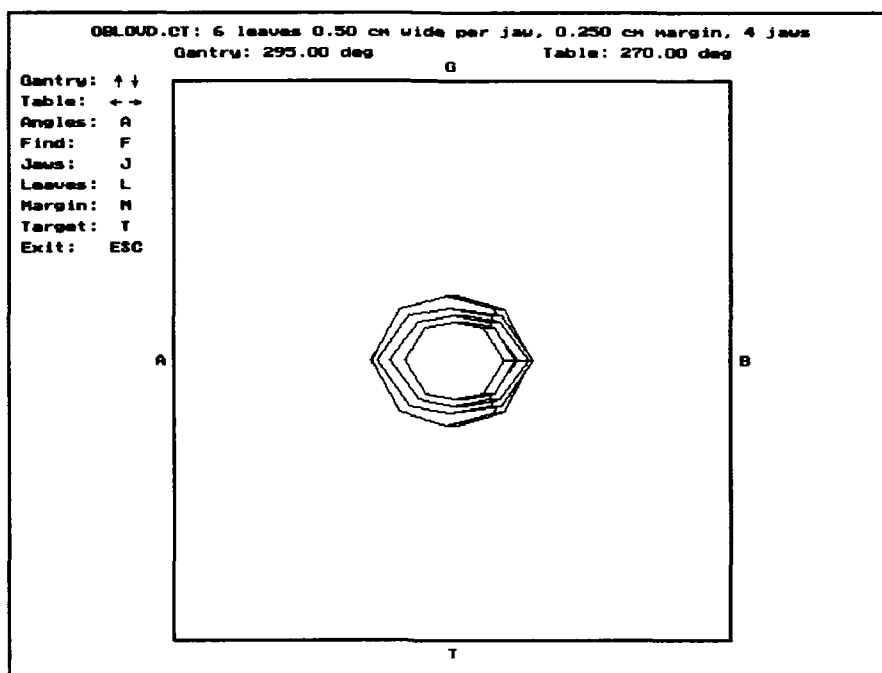


(a)

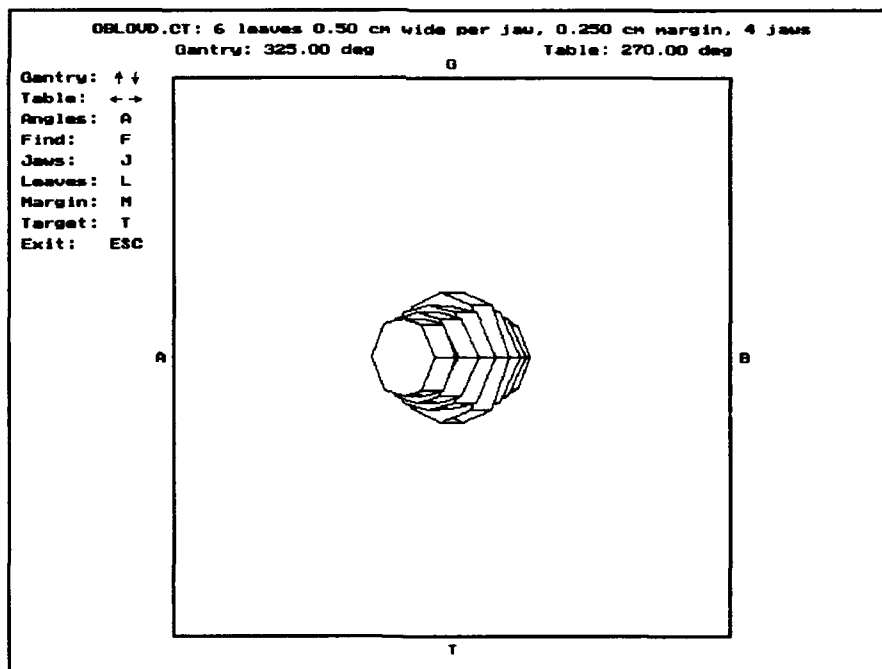


(b)

Figure 4-7: Oblique ovoid target  
 (a) Gantry 235°; (b) Gantry 265°

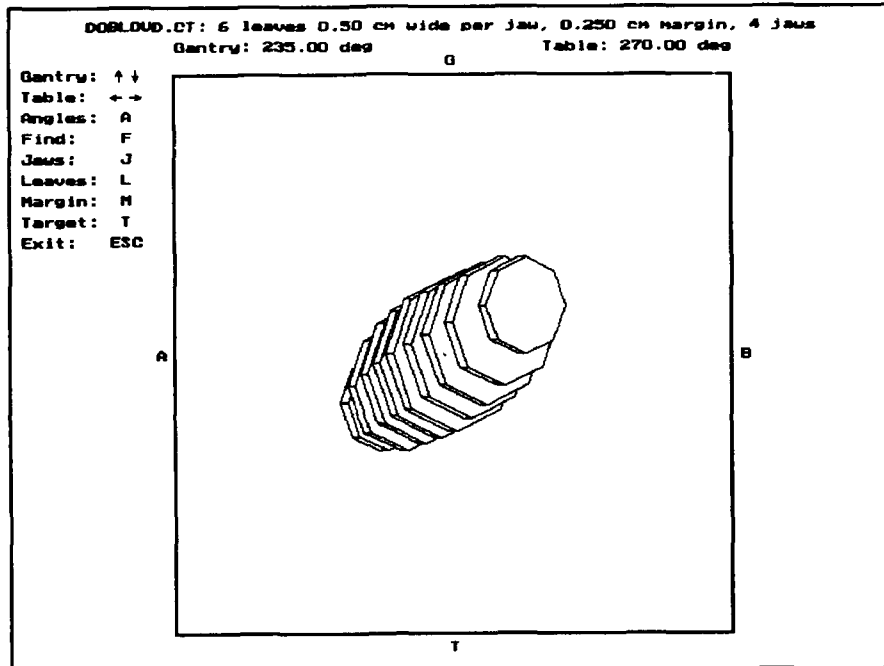


(c)

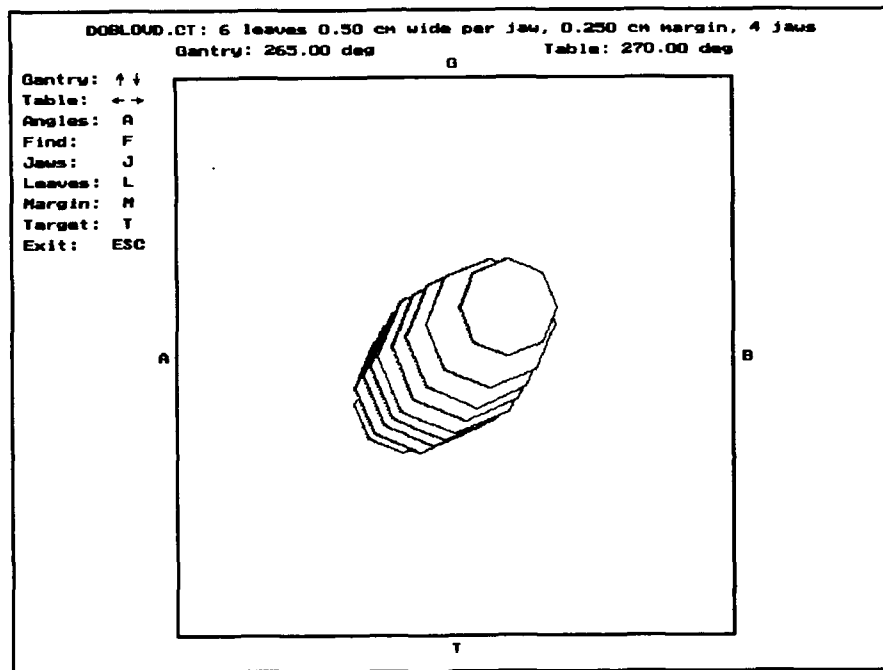


(d)

Figure 4-7 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

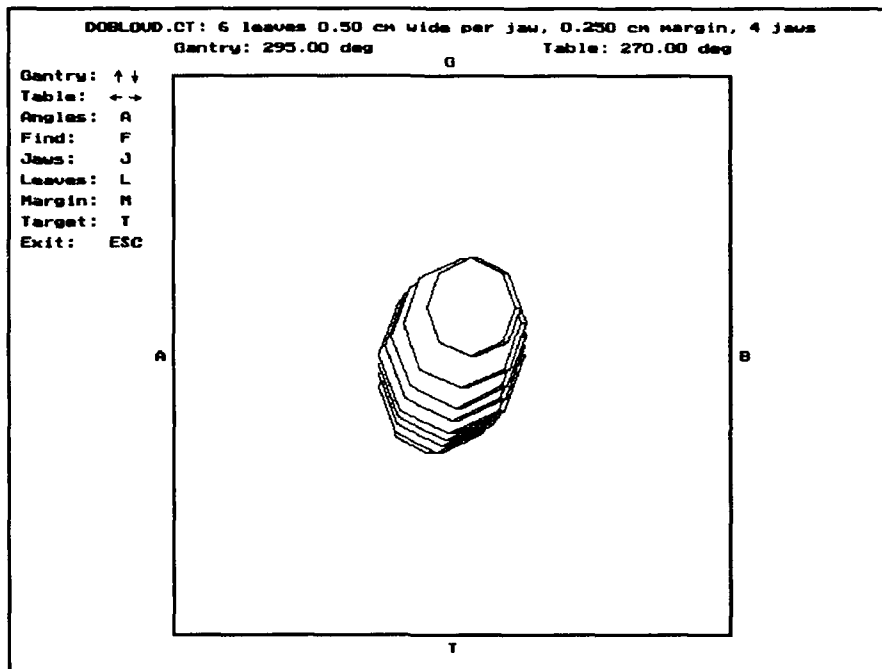


(a)

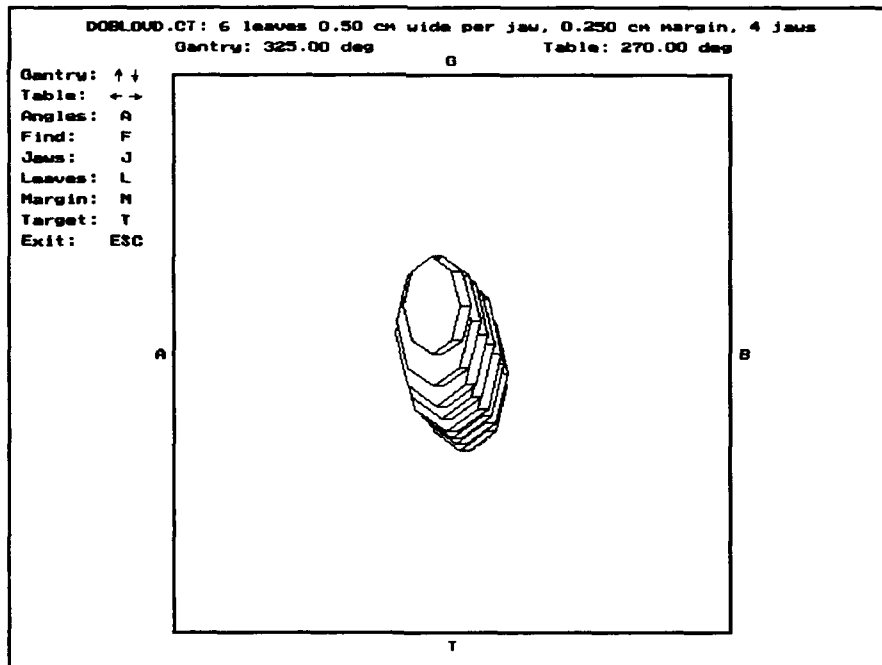


(b)

Figure 4-8: Doubly oblique ovoid target  
 (a) Gantry 235°; (b) Gantry 265°



(c)



(d)

Figure 4-8 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

superior to posterior inferior. The doubly oblique ovoid is similar, but with its major axis oriented from left anterior superior to right inferior posterior. The figures show a series of rotated beam's eye views of the target at a table angle of  $270^\circ$  with gantry angles of  $235^\circ$ ,  $265^\circ$ ,  $295^\circ$ , and  $325^\circ$  (covering a standard  $100^\circ$  arc in four steps). The axial CT points have been tiled to a surface by extending the points to plus and minus one-half of the slice thickness and connecting the related points to form a series of stacked right prisms. The resulting structure is then submitted to the rotation algorithm. The final rotated images at each gantry position show the appropriately scaled (rotation center at 100 cm, view screen at 70 cm) projected area of the target. Determination of the boundary of this projected area is then necessary to correctly position the leaves of the multileaf collimator.

#### Target Localization

The target is localized by a stepwise graphical search method. Consider the individual elements (pixels) of each graphics vector in the target representation to be in a set state. Those elements that are set on the periphery define the projected cross section of the target, suitably scaled to viewing distance. As these peripheral elements are the sole elements of interest, hidden line removal in the representation of the target in the localization program is unnecessary.



The leading, or field, edge of each leaf, in turn clockwise from the upper left in the BEV coordinate system (the gantry leaf on the A side), is advanced by one element. Each element on the leading edge of the leaf is then sequentially scanned to determine if coincidence with a set element has occurred. If no set elements are found, the sides of the leaf are checked by rotating about each apex at the margin radius, as illustrated in figure 4-9.

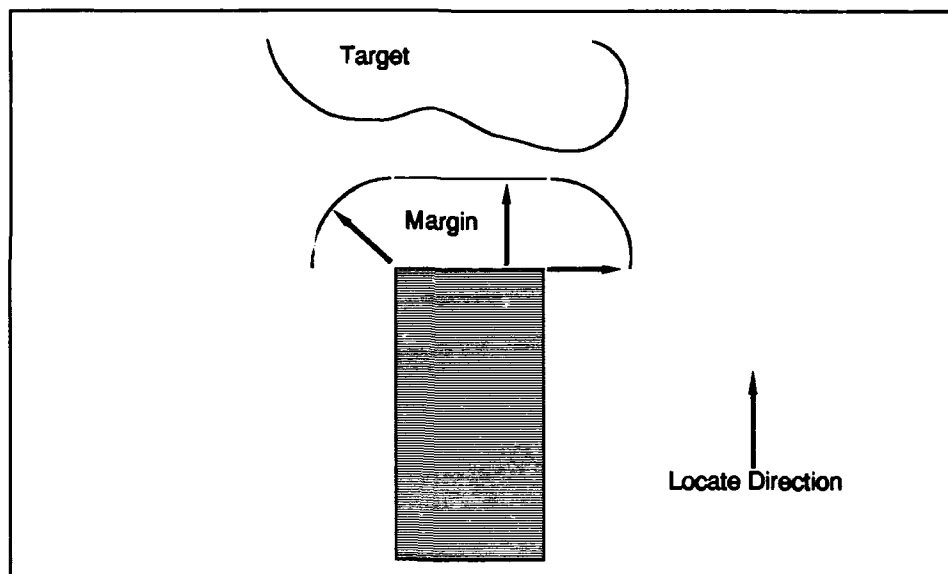
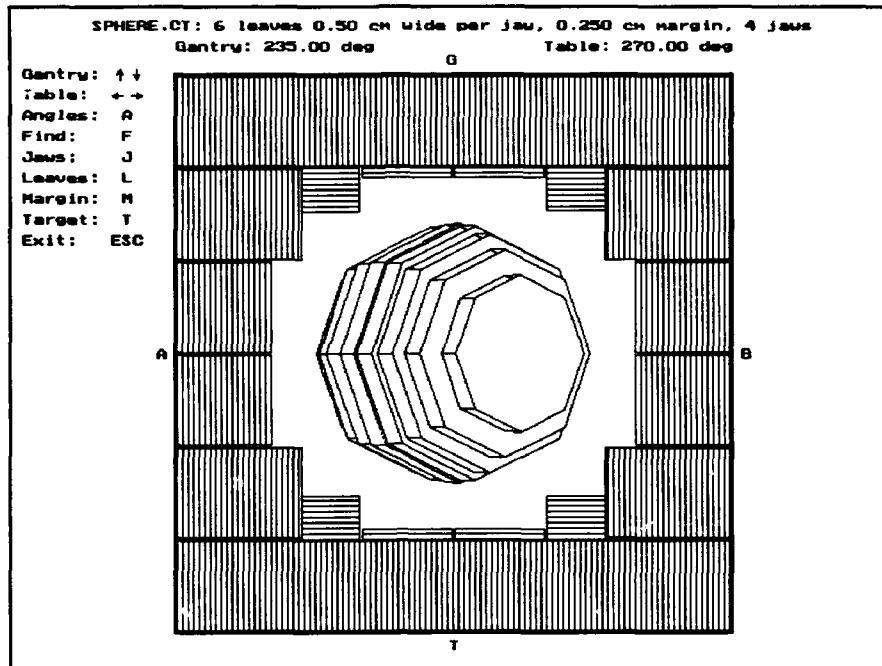


Figure 4-9: Target search and localization

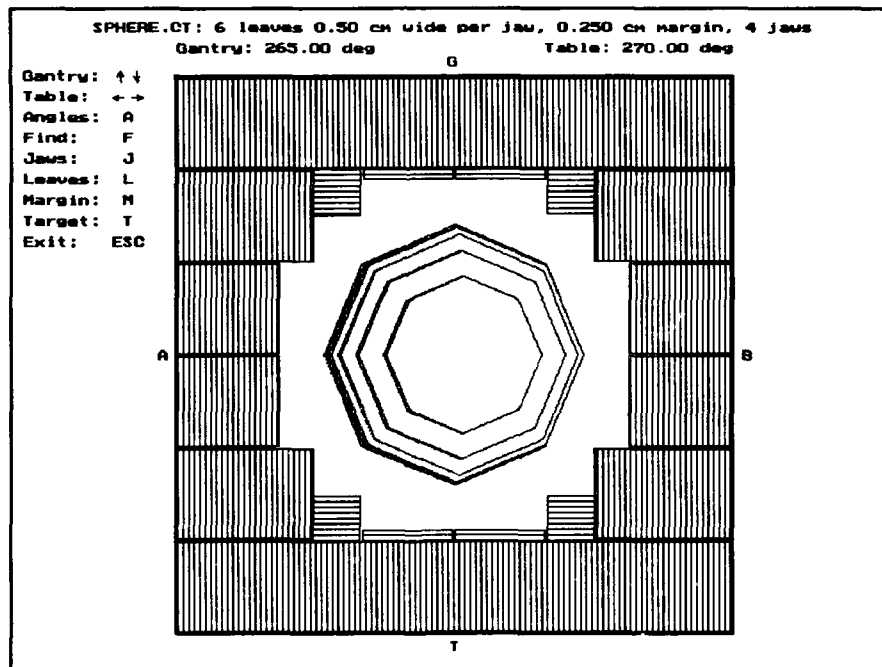
If a set element is not encountered to the sides, the leaf edge is again advanced one element and scanned. This process continues until a set element is encountered (either the target periphery, the edge of an opposing leaf where applicable, or the limit of the view window which defines the limit of the collimator open aperture). Leaf movement is stopped, the leaf position is translated from screen

coordinates to world coordinates, and the position is recorded. After all leaf positions have been resolved, the settings are sent to a data file for processing by the dosimetry program.

The result of the localization is shown in figures 4-10 through 4-21 for both four jaw and two jaw multileaf collimators. The targets are the same as in figures 4-3 through 4-8 respectively, as are the table and gantry positions. As many of the multileaf collimators described in the literature are of the two jaw type, a comparison of coverage was deemed appropriate.



(a)



(b)

Figure 4-10: Sphere, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

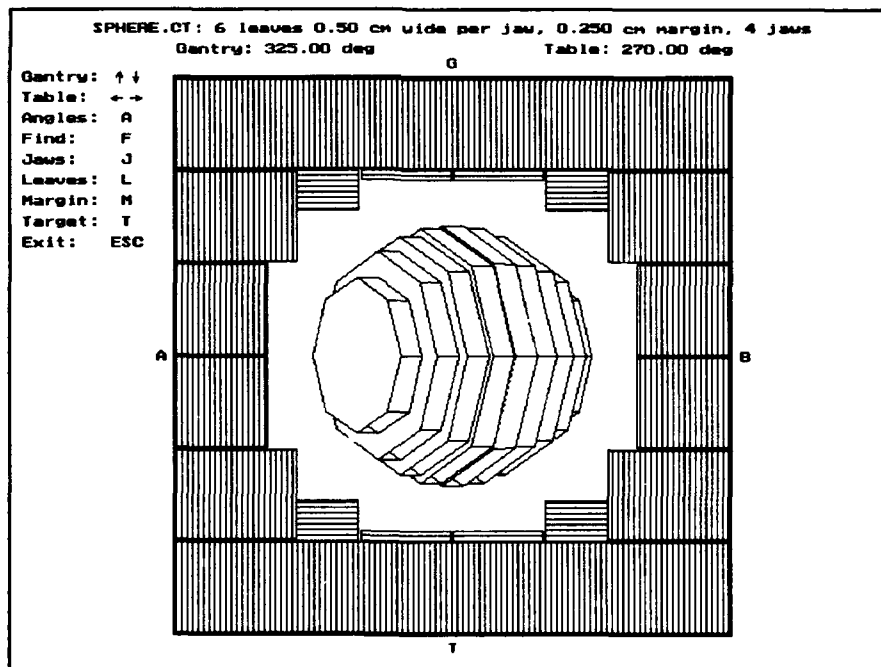
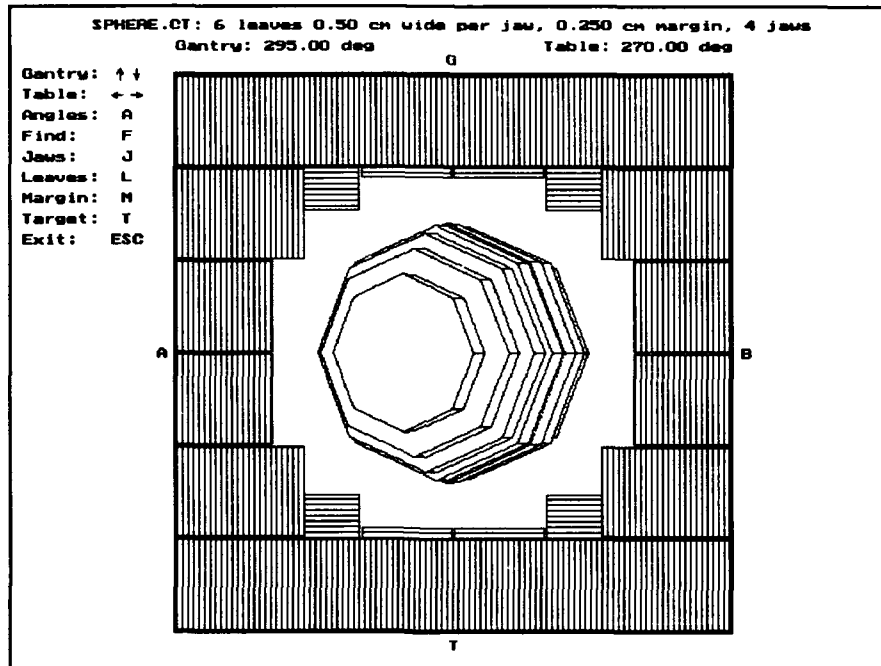
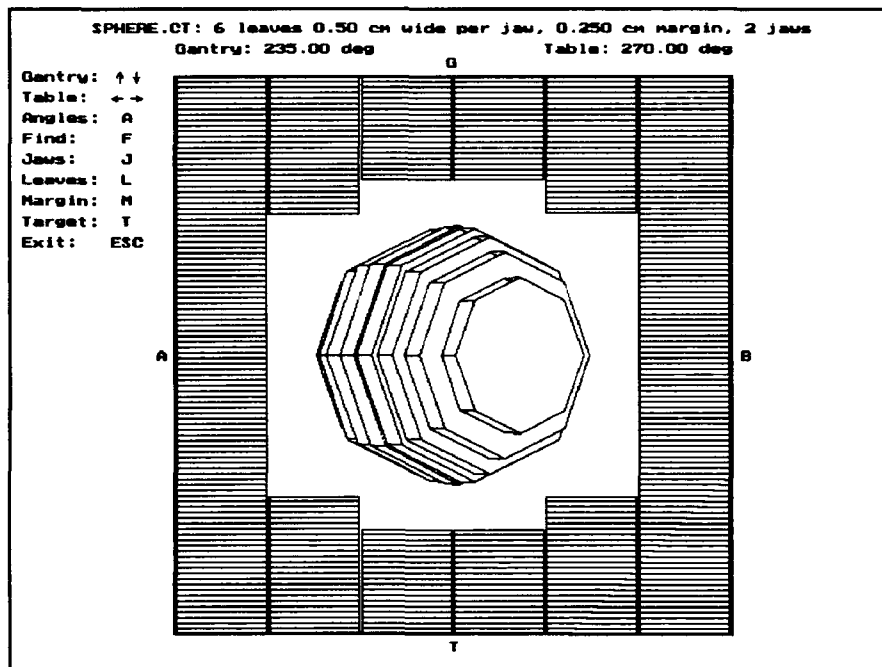
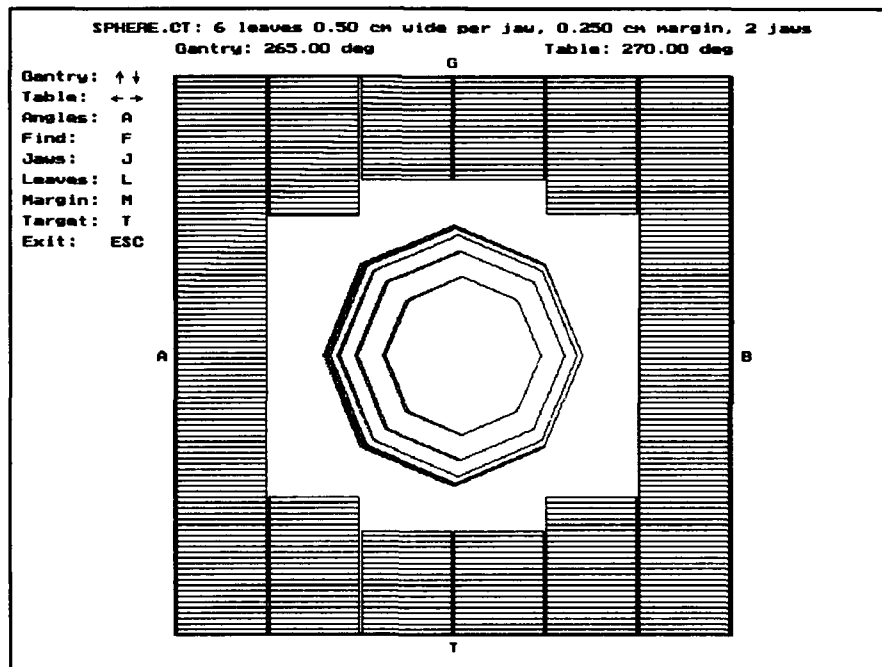


Figure 4-10 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

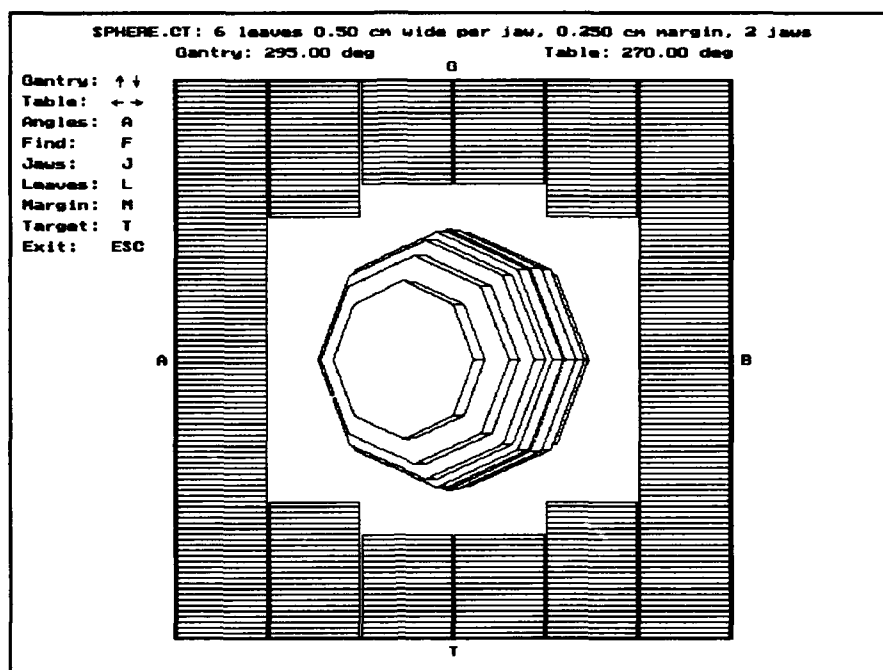


(a)

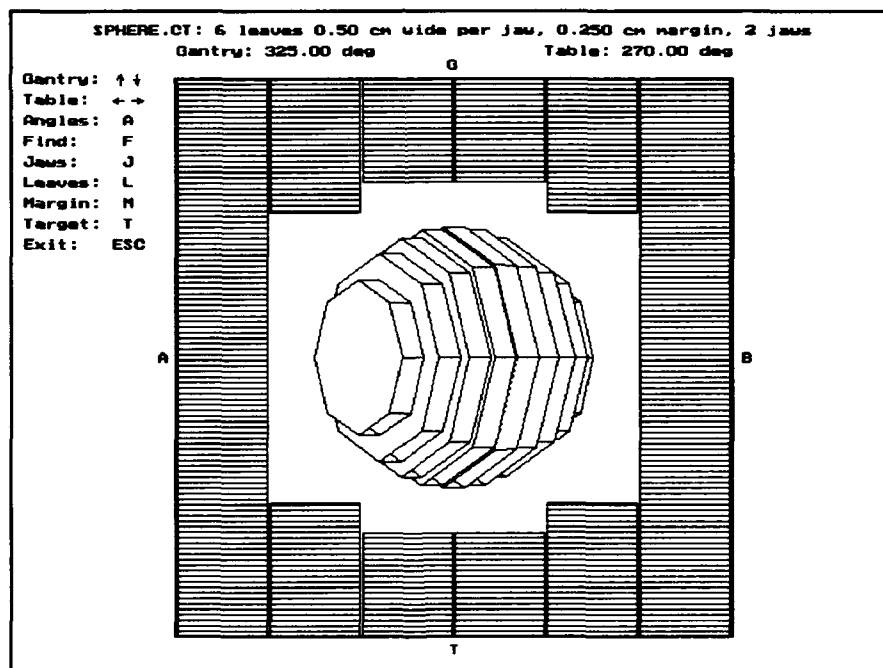


(b)

Figure 4-11: Sphere, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

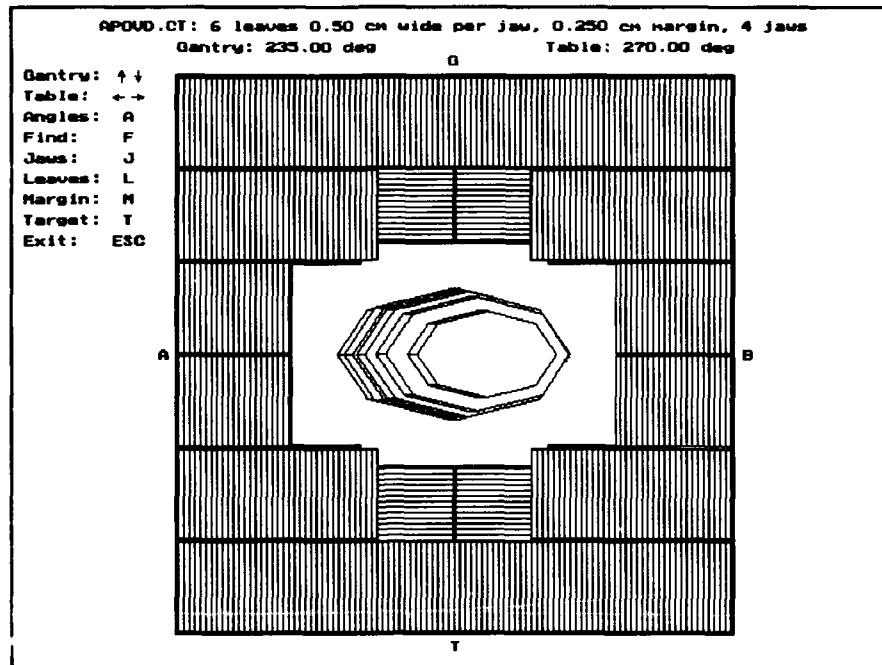


(c)

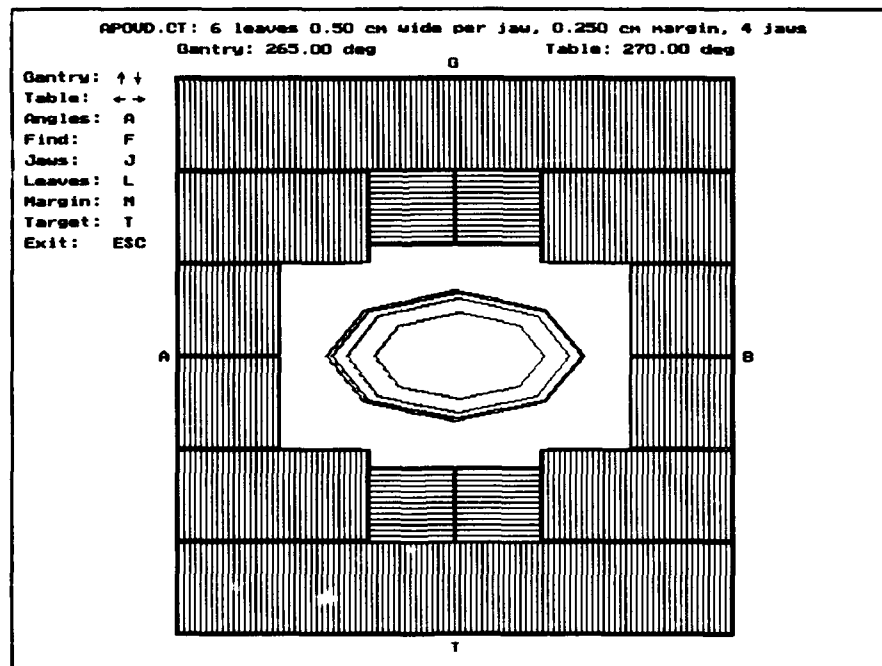


(d)

Figure 4-11 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



(a)



(b)

Figure 4-12: AP ovoid, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

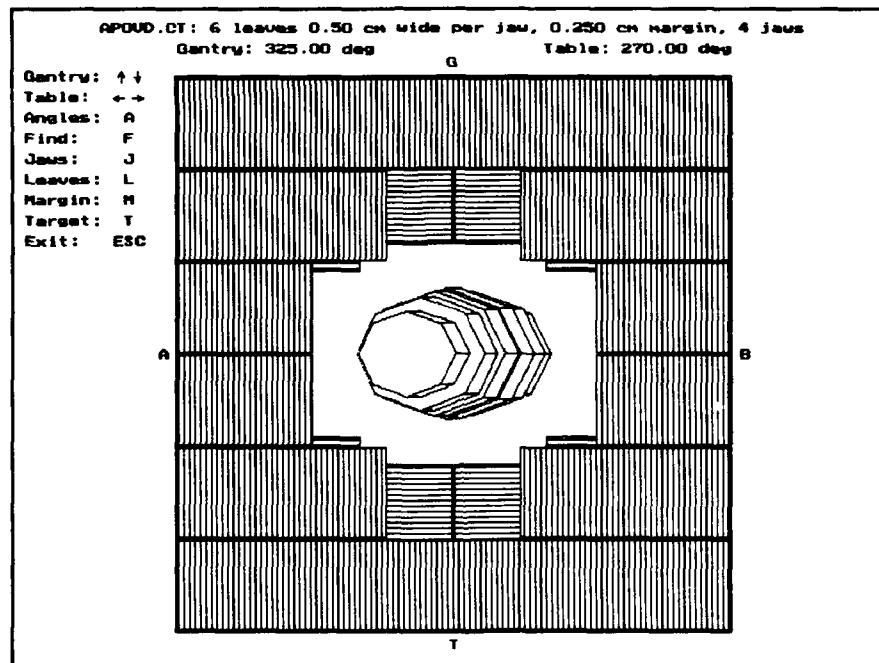
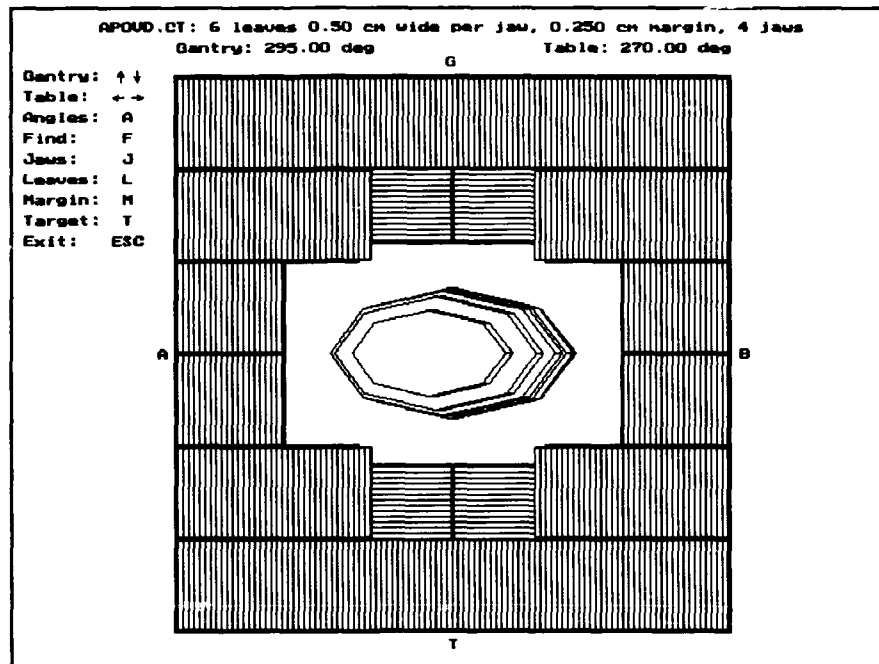
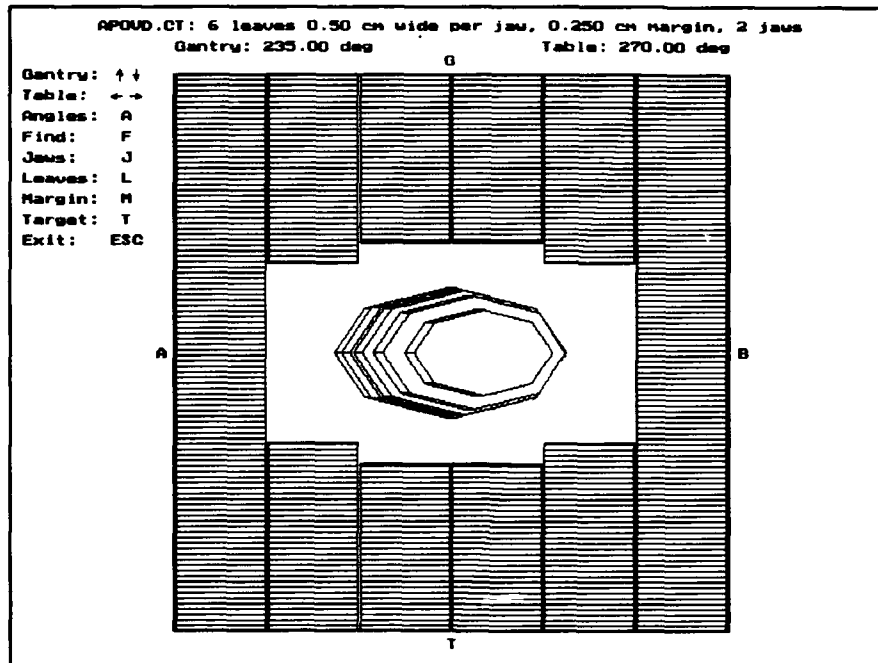
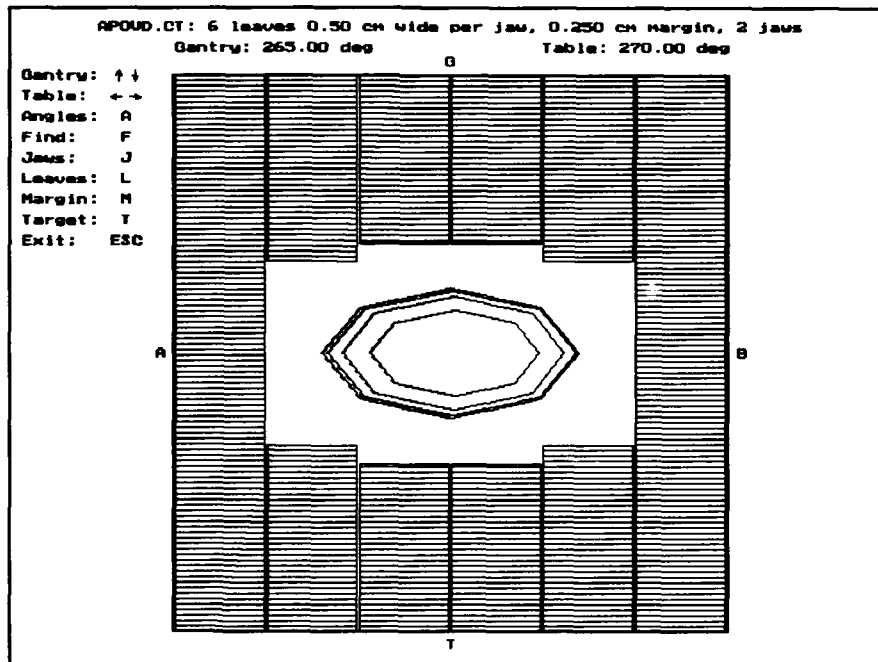


Figure 4-12 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



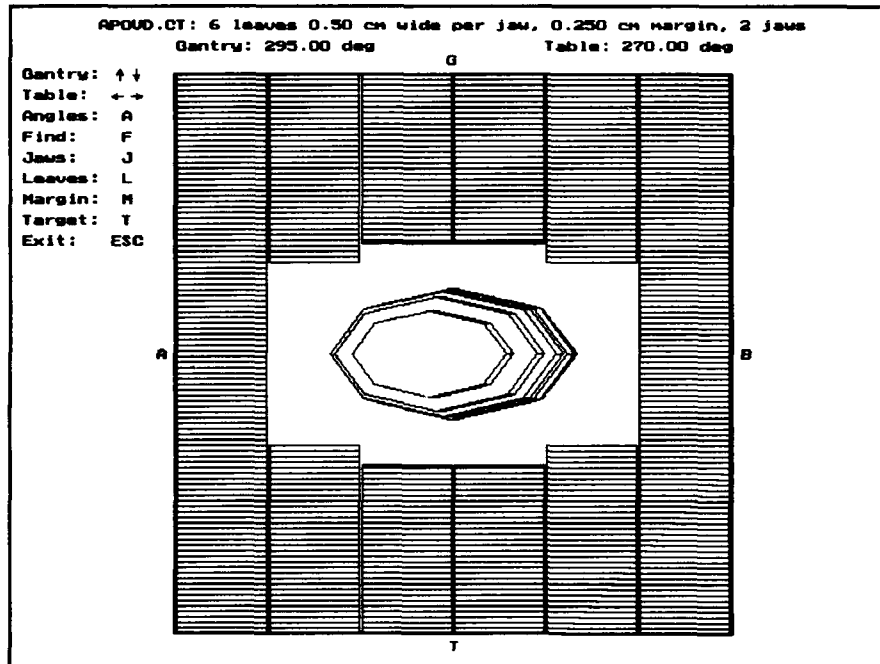


(a)

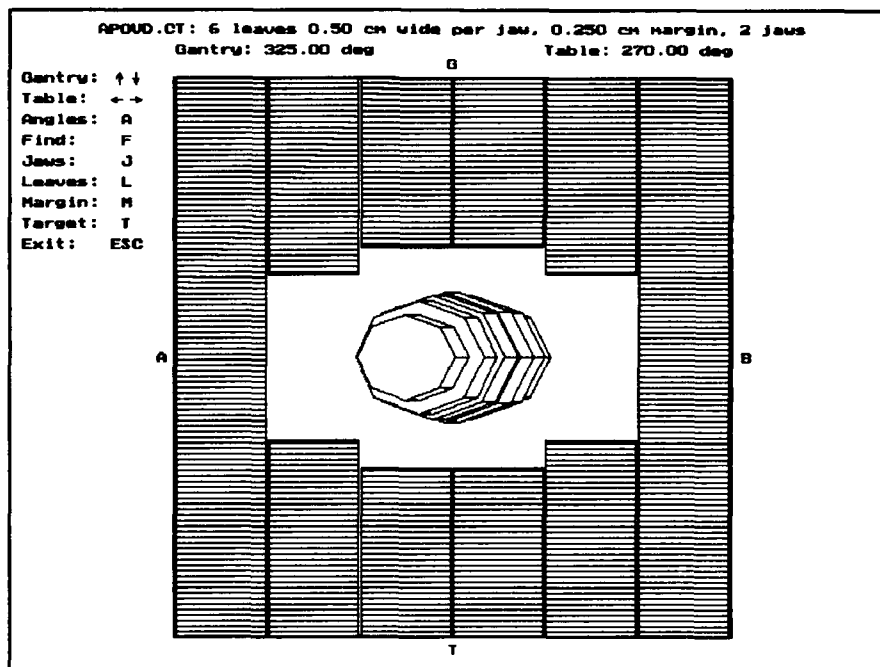


(b)

Figure 4-13: AP ovoid, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

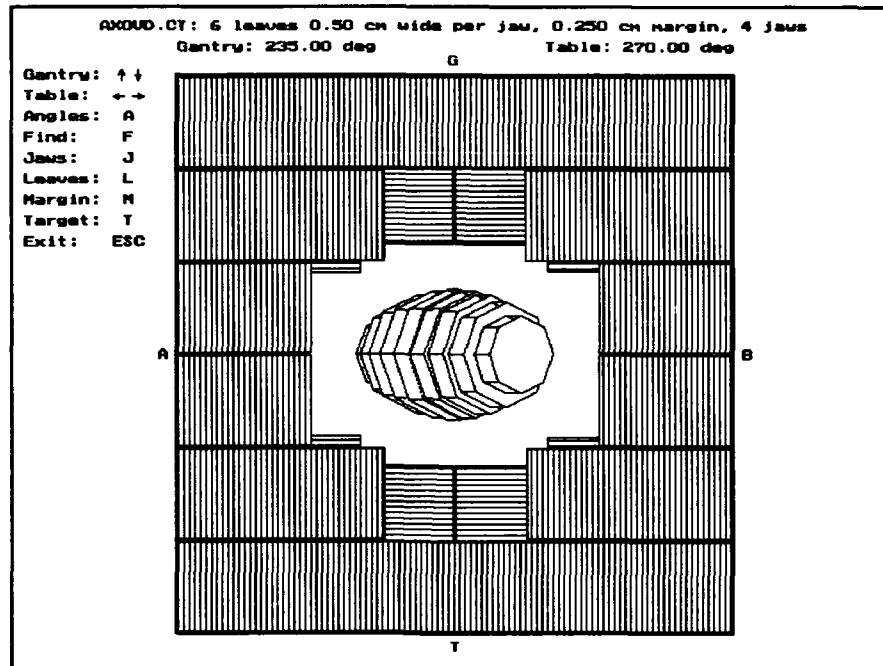


(c)

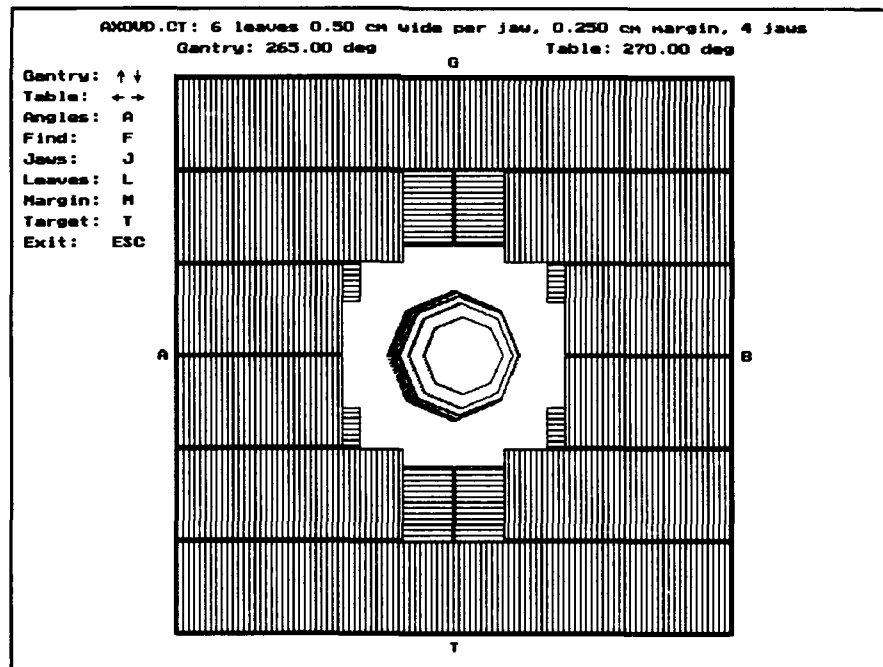


(d)

Figure 4-13 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

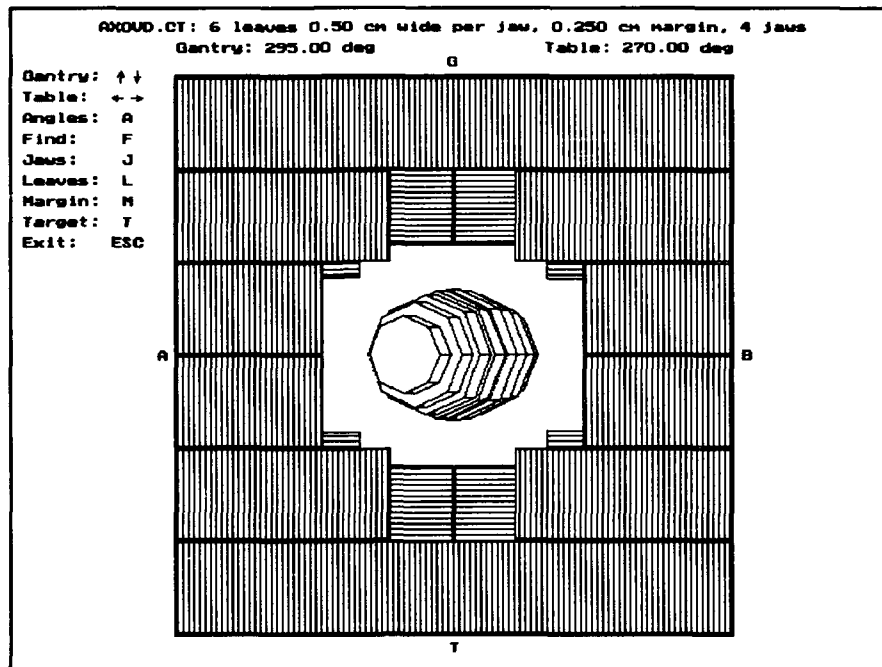


(a)

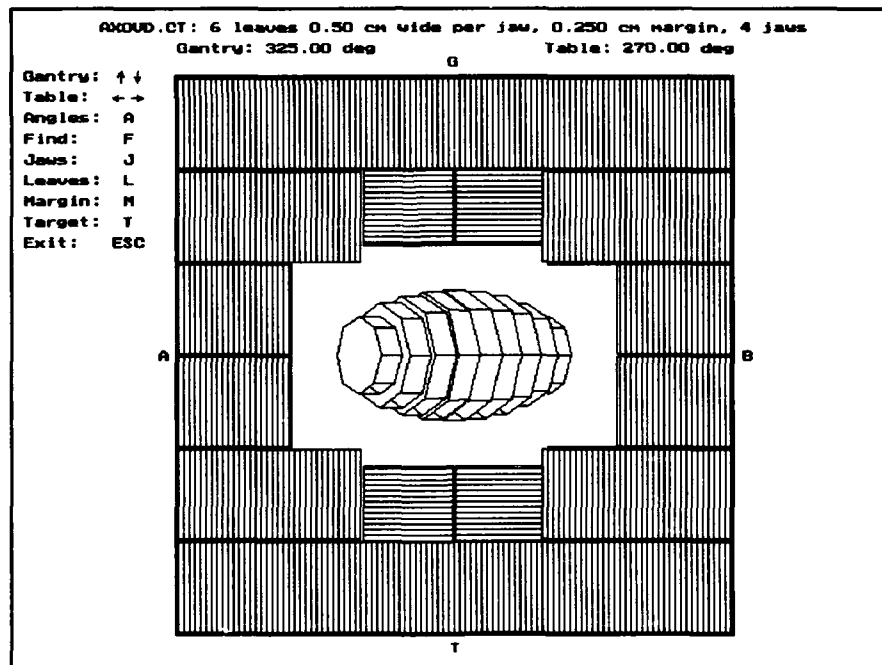


(b)

Figure 4-14: Axial ovoid, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

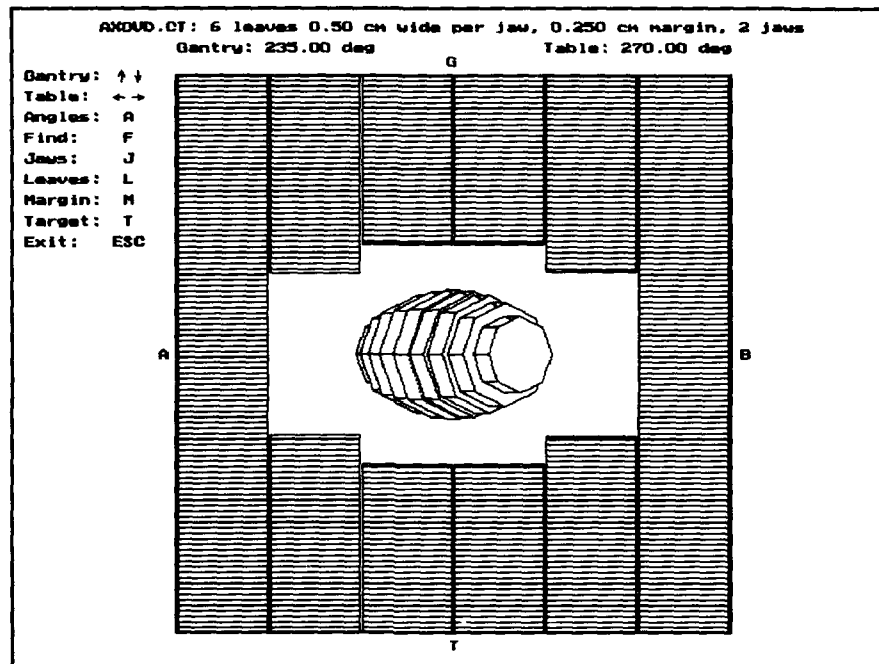


(c)

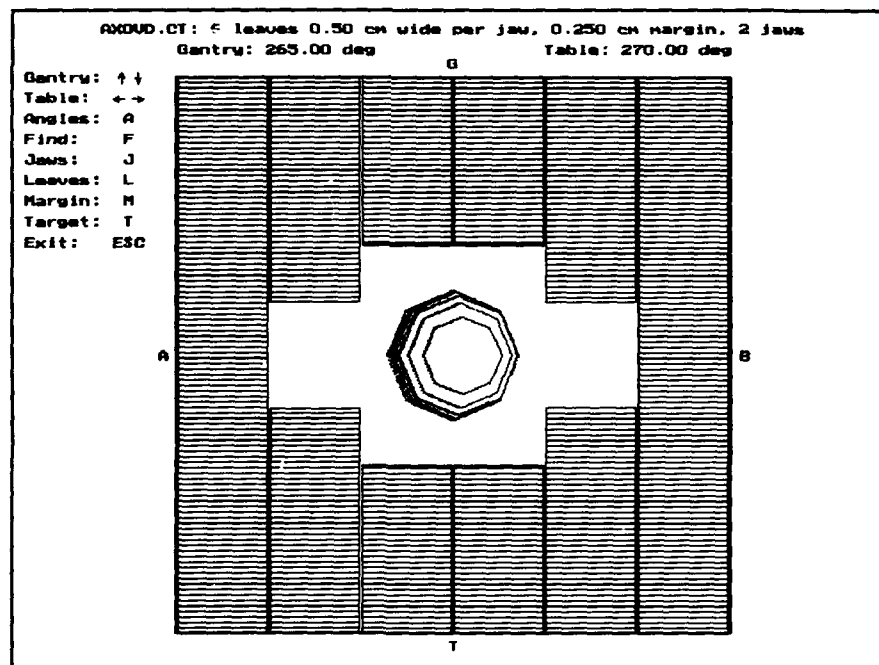


(d)

Figure 4-14 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

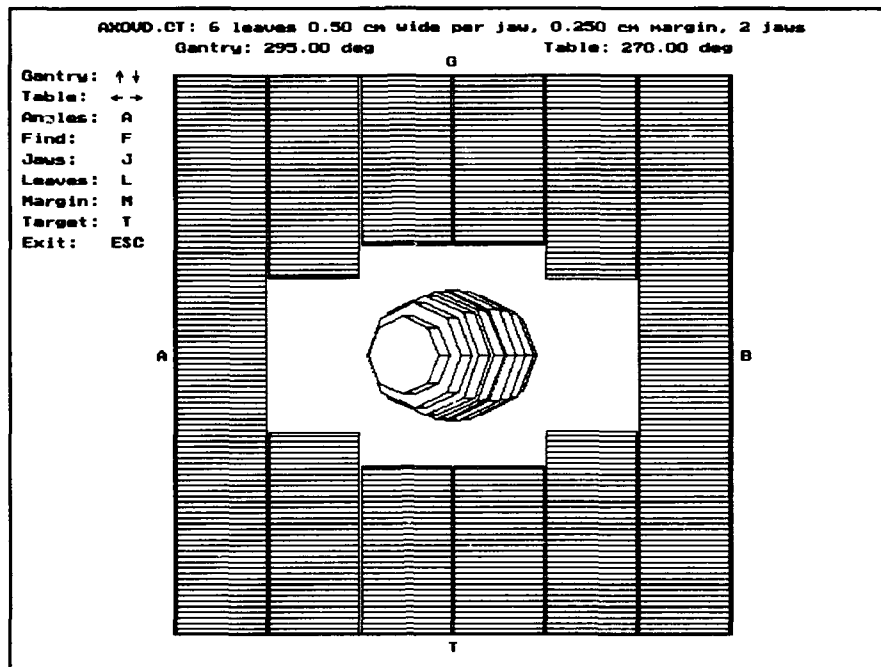


(a)

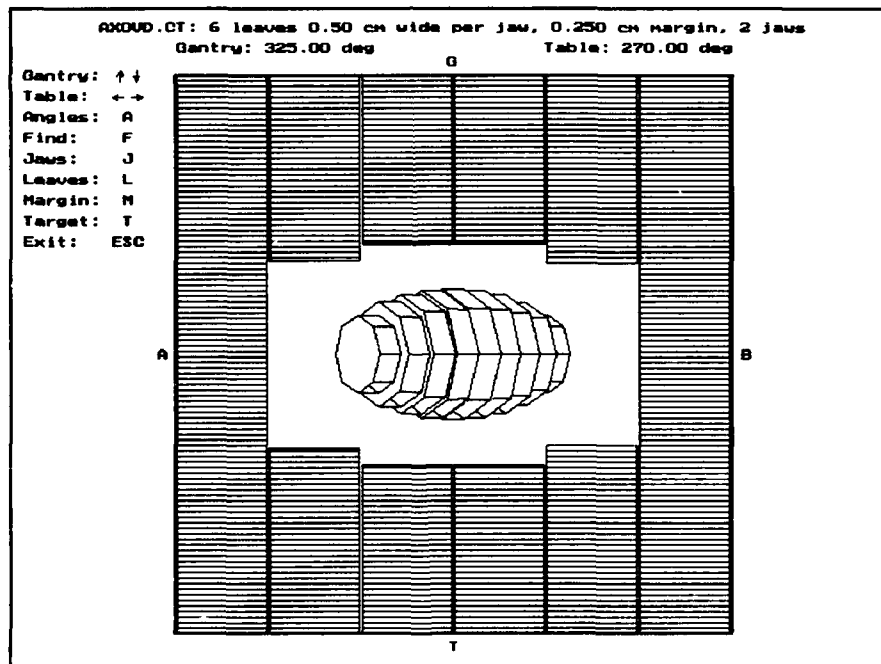


(b)

Figure 4-15: Axial ovoid, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

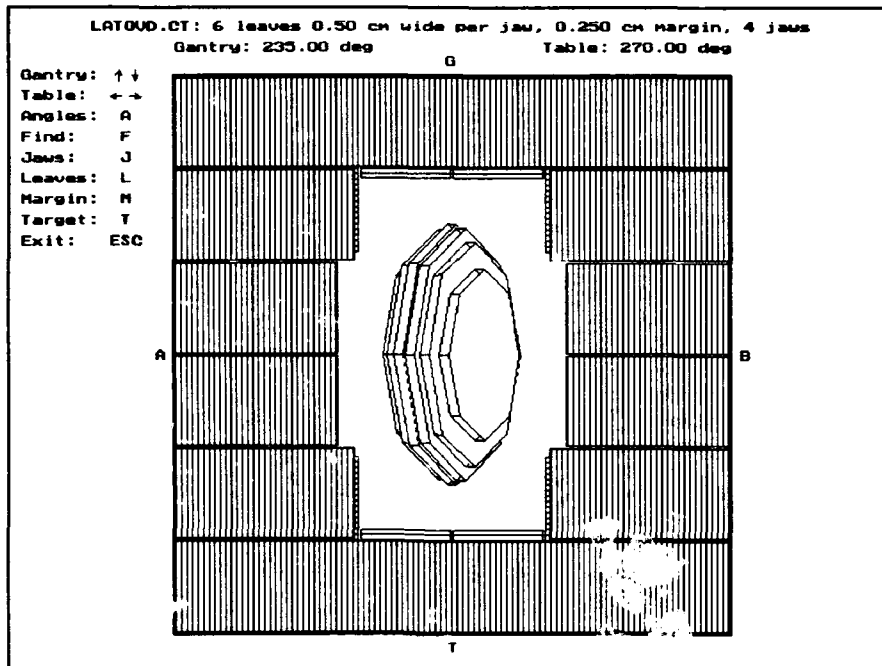


(c)

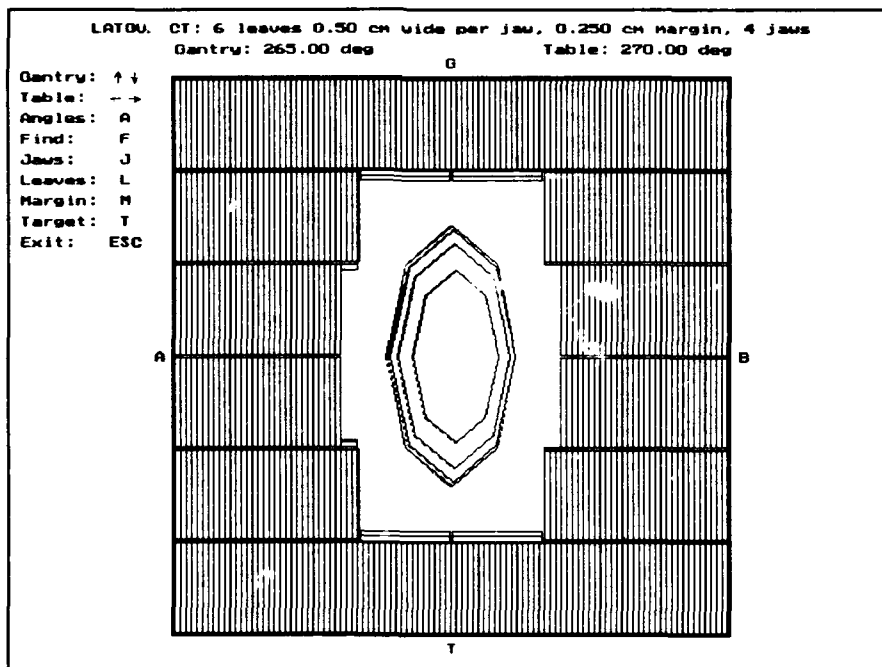


(d)

Figure 4-15 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

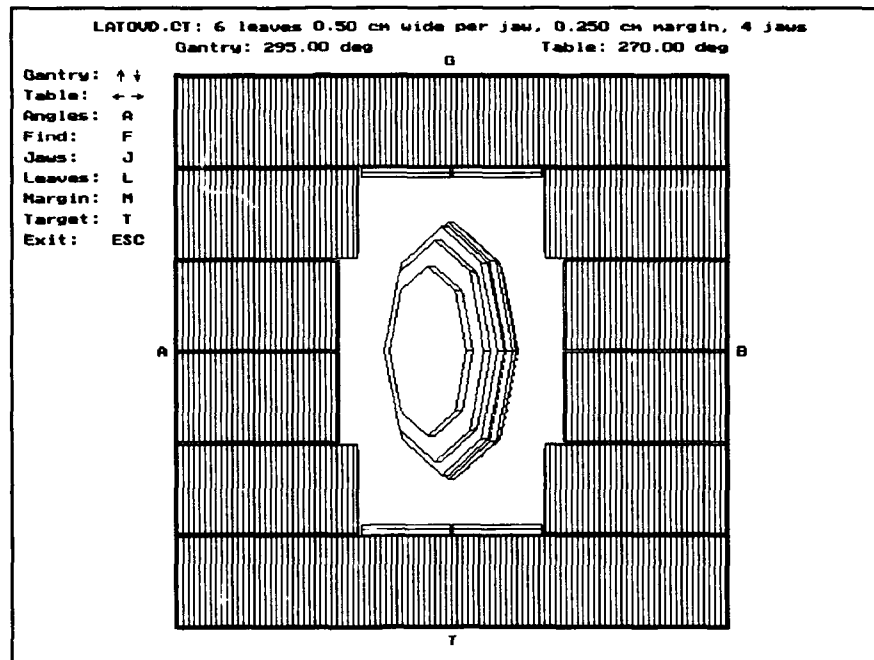


(a)

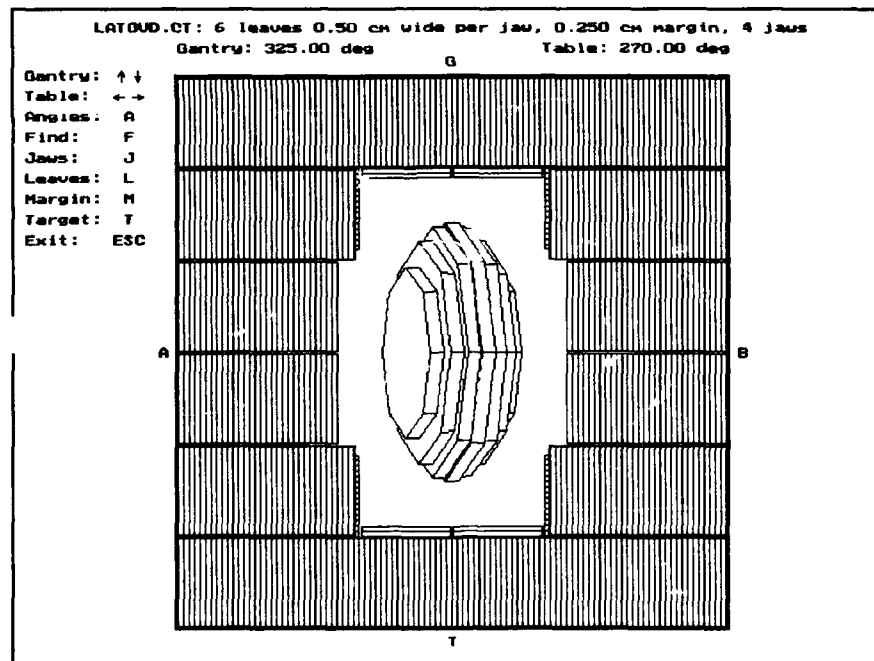


(b)

Figure 4-16: Lateral ovoid, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°



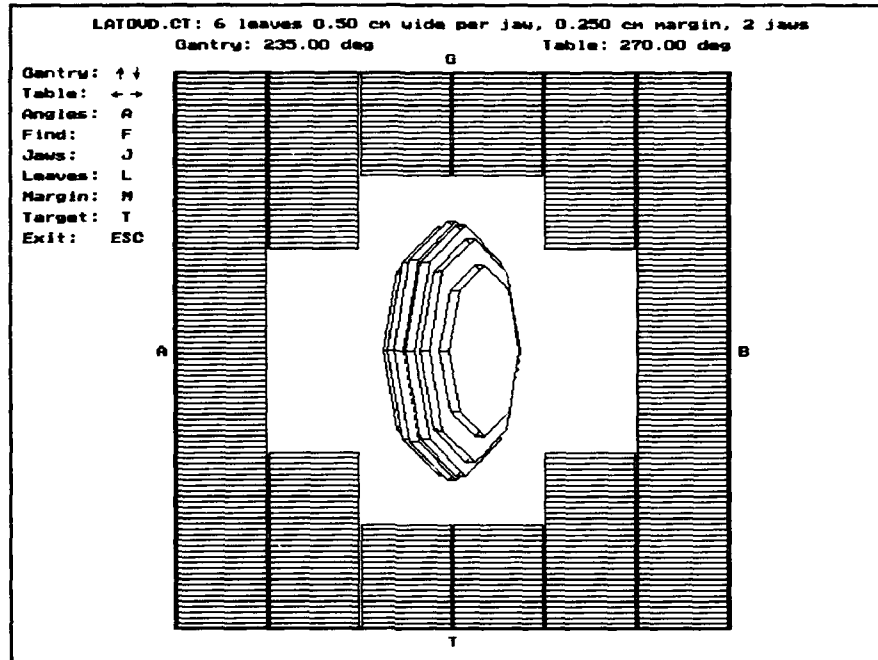
(c)



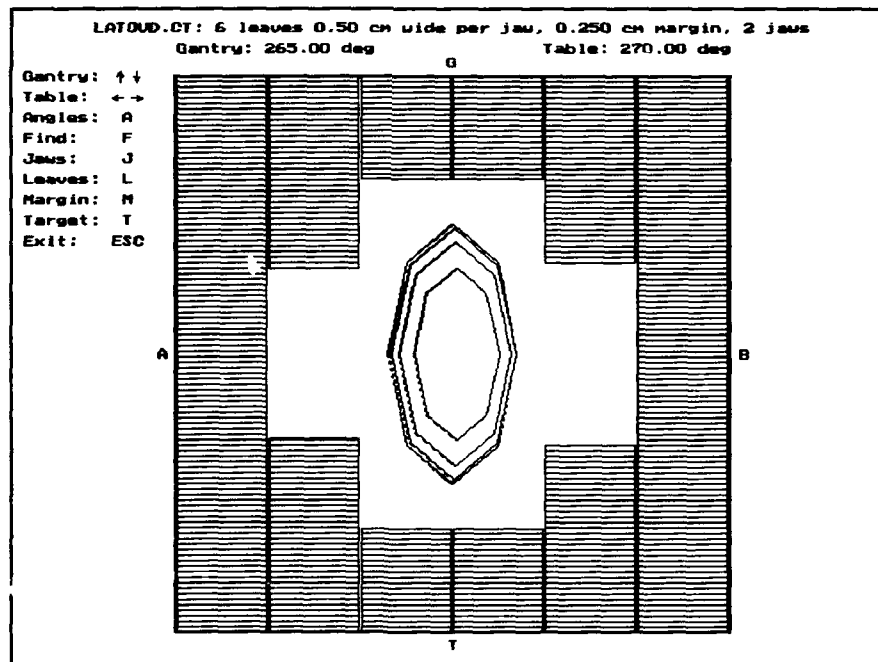
(d)

Figure 4-16 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



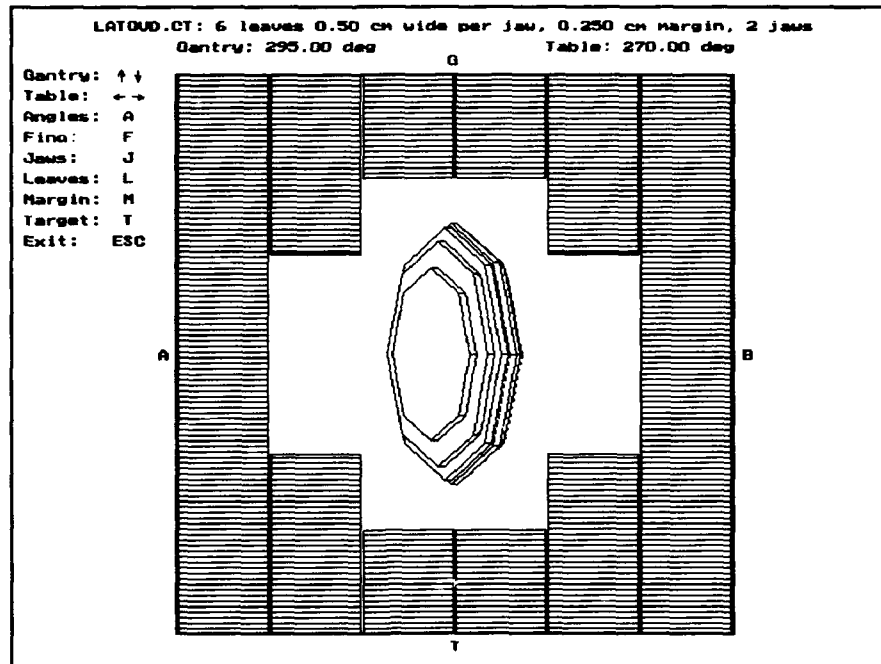


(a)

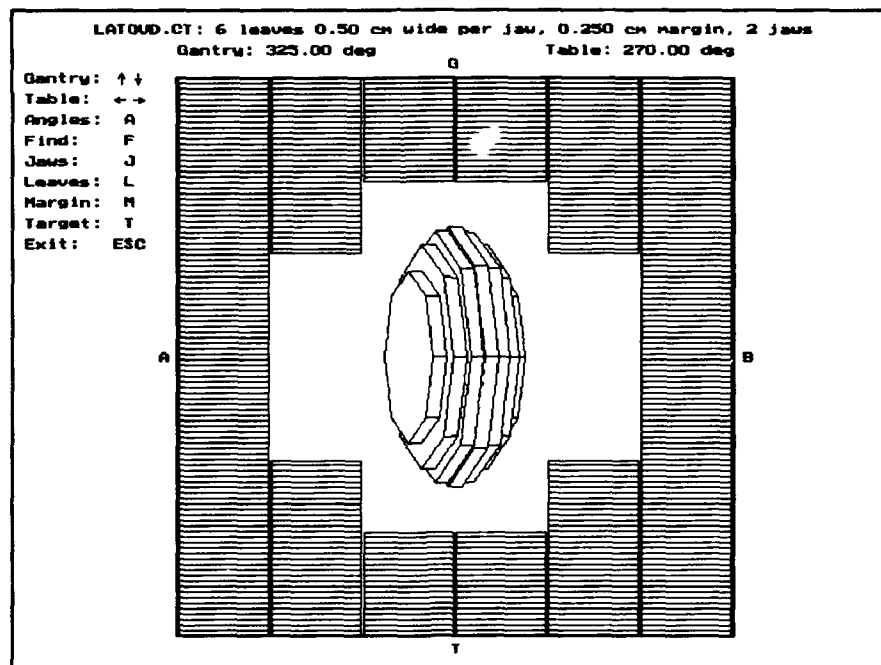


(b)

Figure 4-17: Lateral ovoid, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

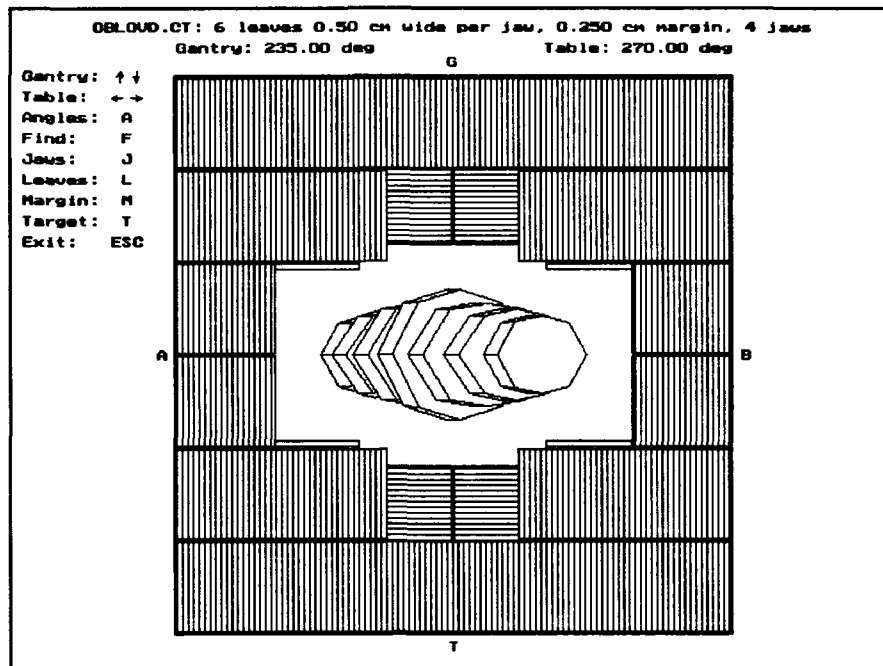


(c)

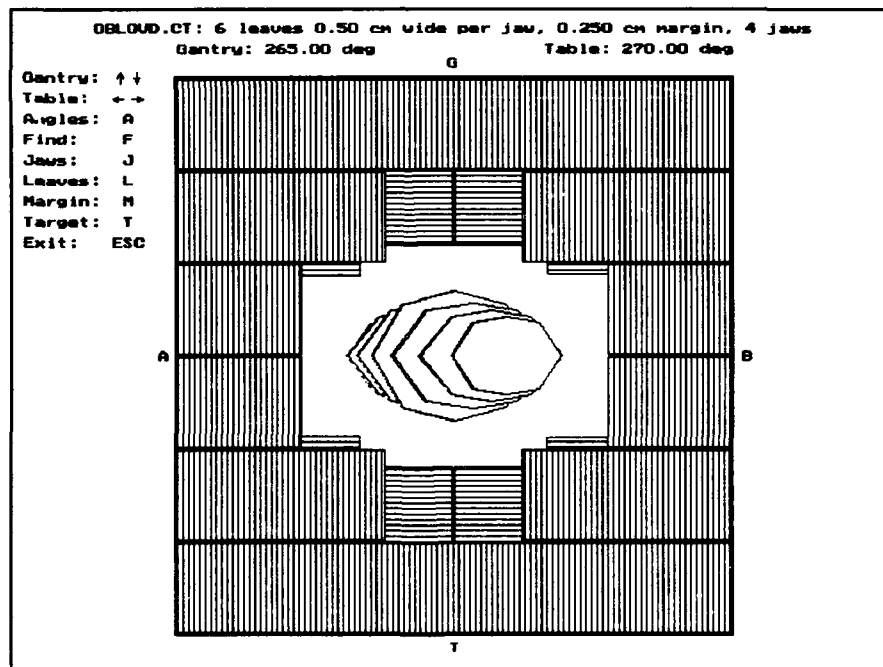


(d)

Figure 4-17 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

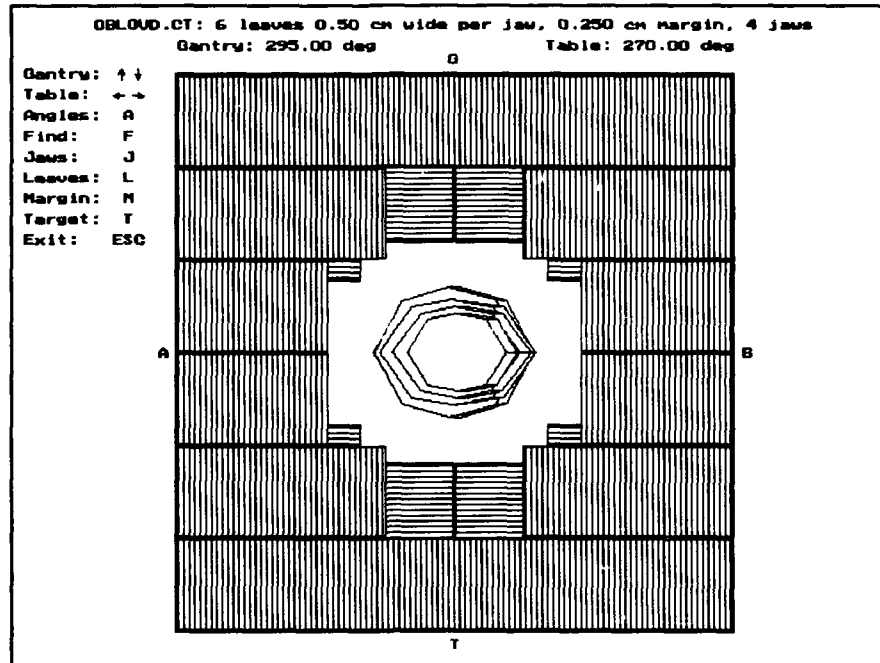


(a)

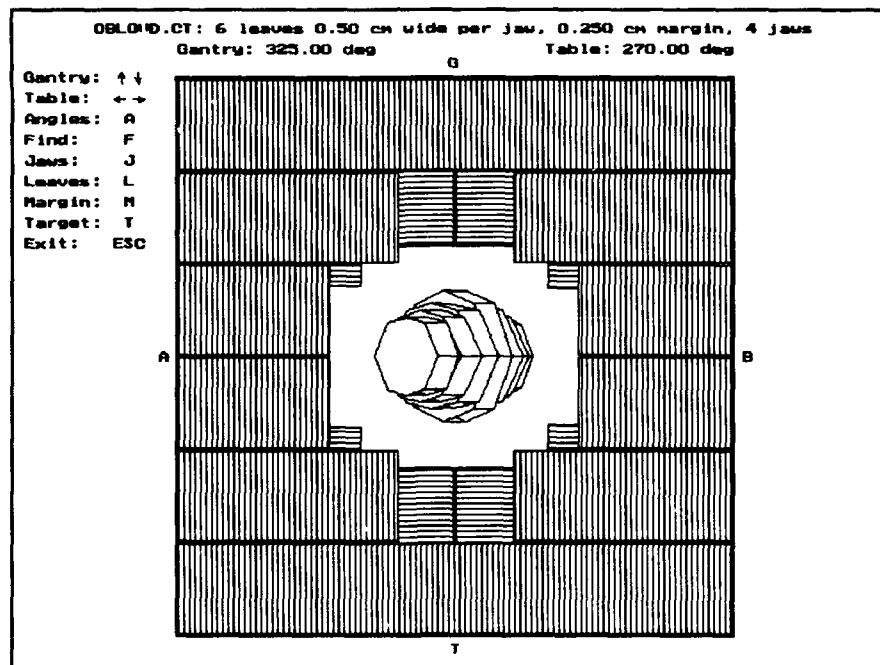


(b)

Figure 4-18: Oblique ovoid, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

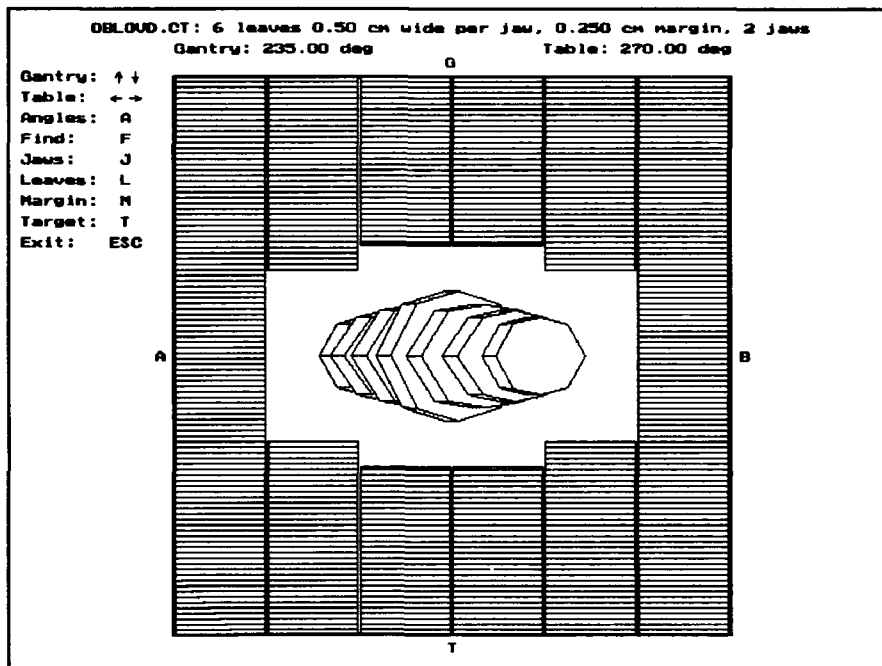


(c)

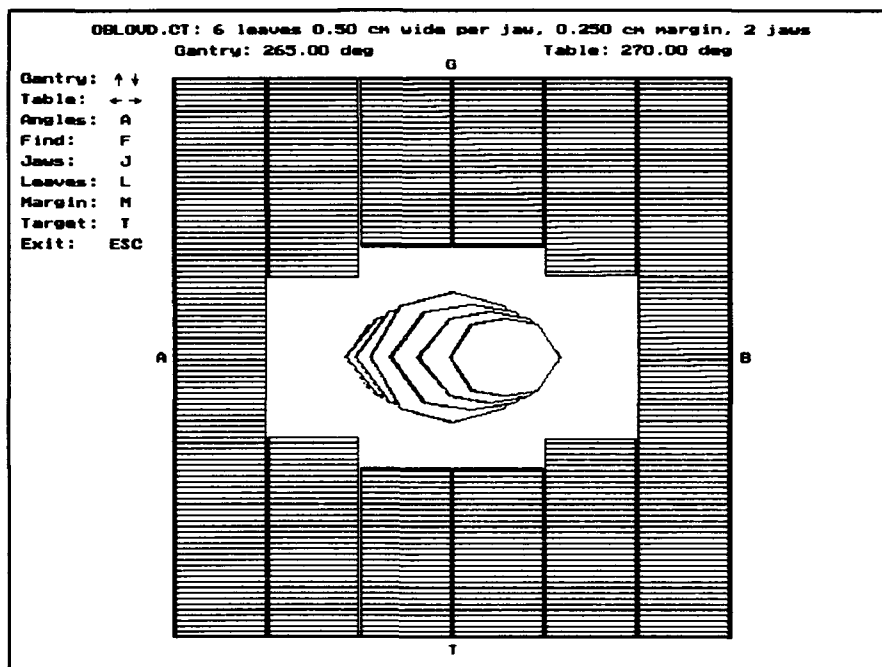


(d)

Figure 4-18 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

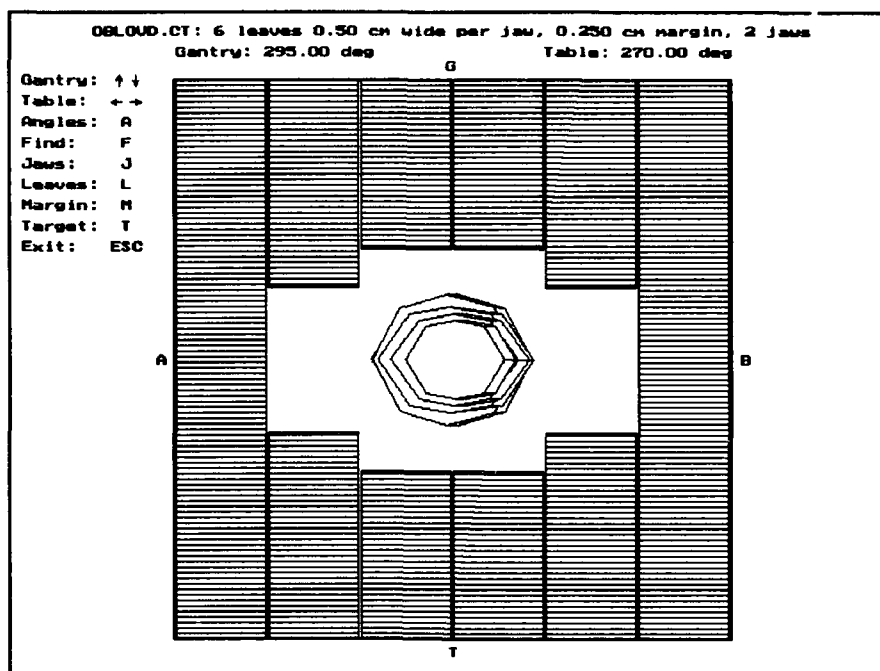


(a)

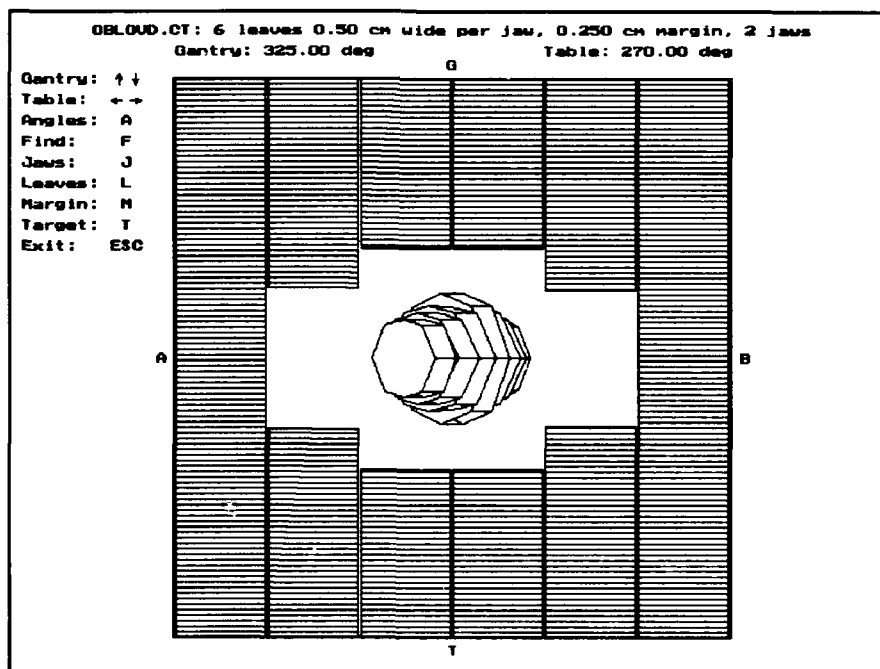


(b)

Figure 4-19: Oblique ovoid, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

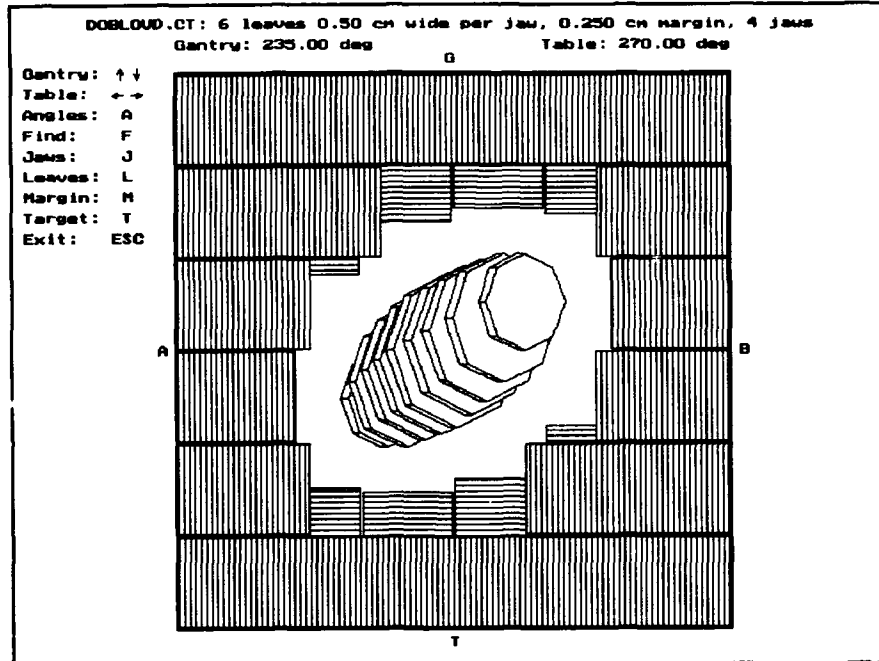


(c)

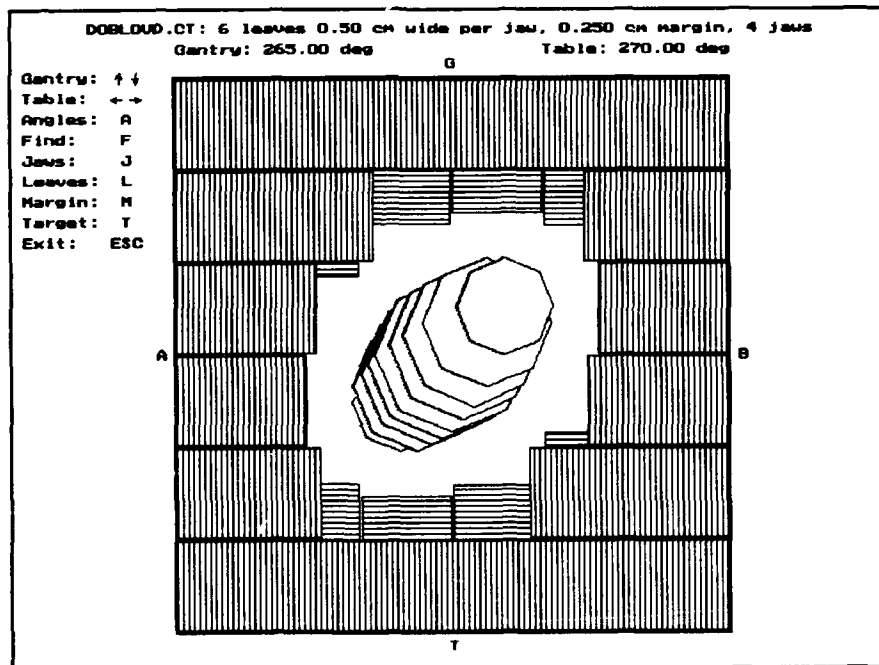


(d)

Figure 4-19 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



(a)



(b)

Figure 4-20: Double oblique ovoid, 4 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°

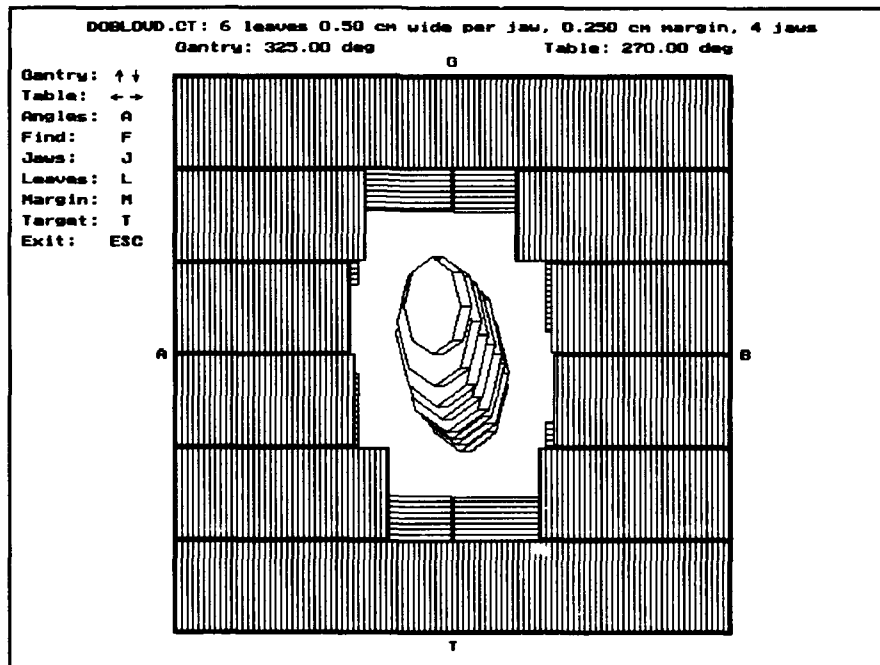
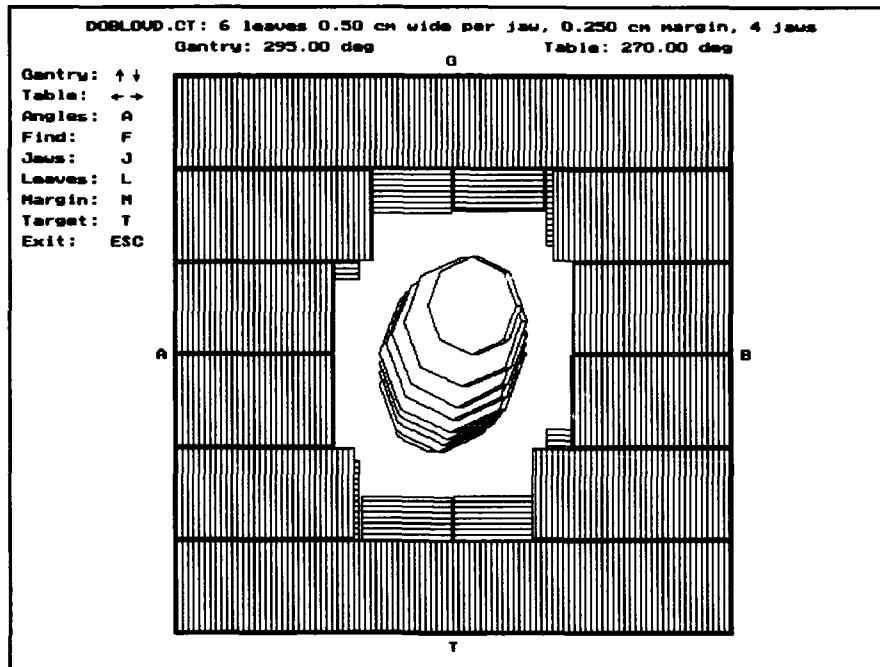
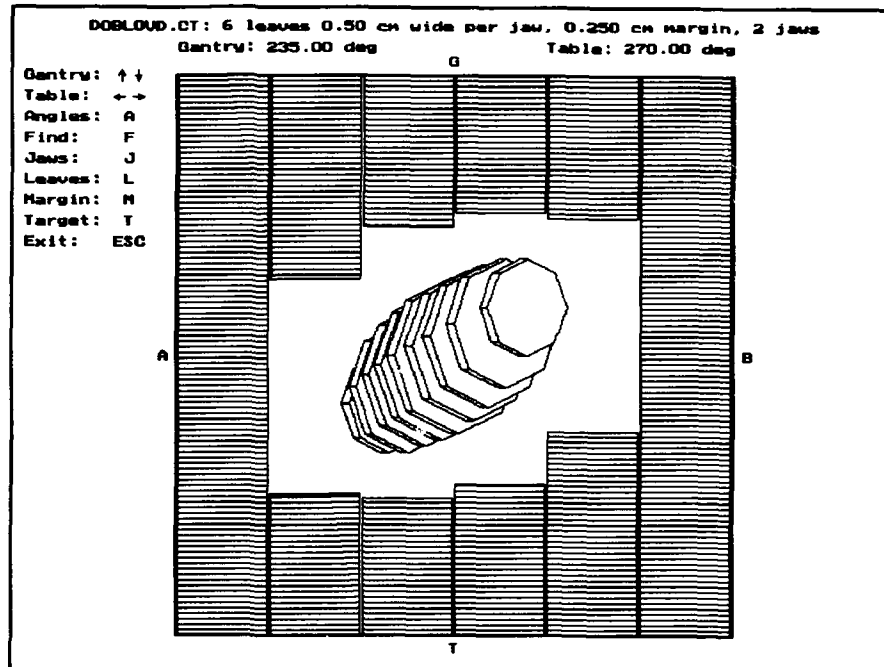
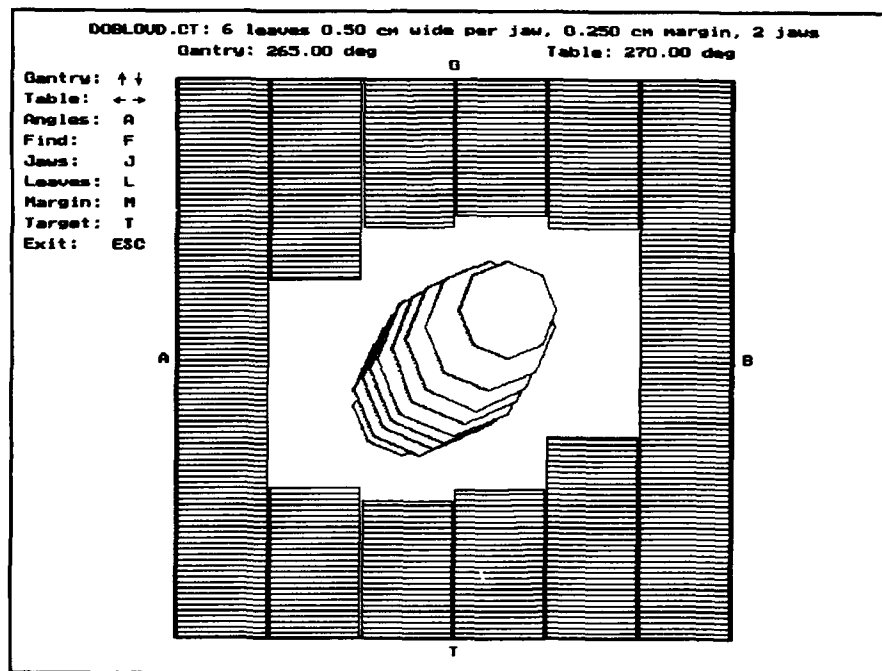


Figure 4-20 -- continued  
 (c) Gantry 295°; (d) Gantry 325°



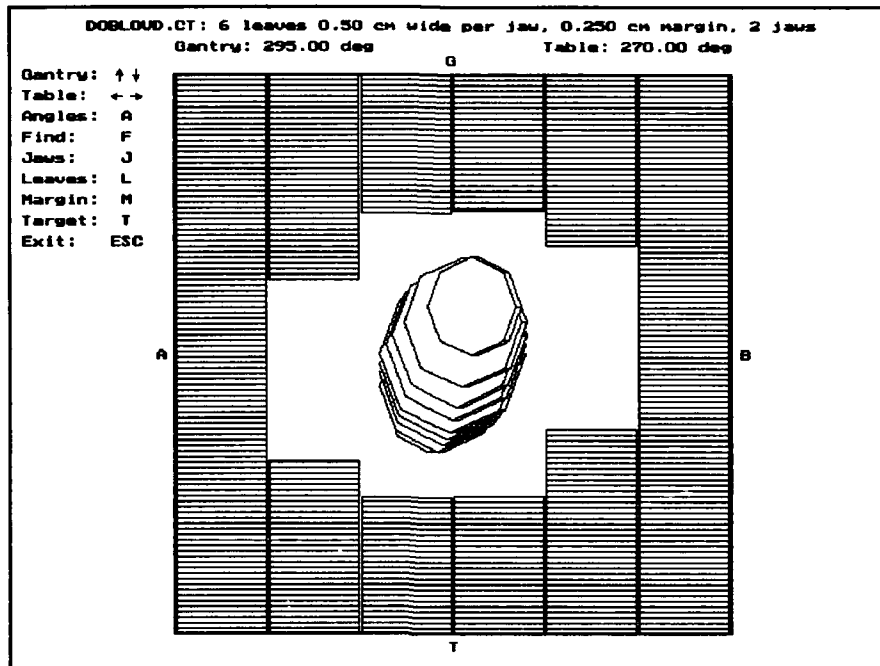


(a)

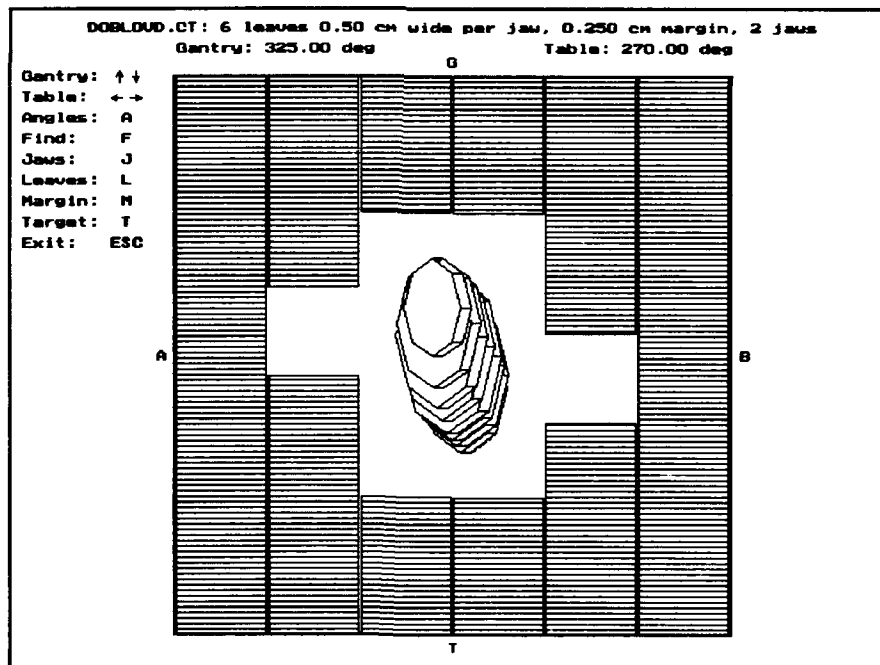


(b)

Figure 4-21: Double oblique ovoid, 2 jaw localized  
 (a) Gantry 235°; (b) Gantry 265°



(c)



(d)

Figure 4-21 -- continued  
 (c) Gantry 295°; (d) Gantry 325°

## CHAPTER 5

### INVESTIGATIONAL DOSIMETRY

The current dose model used in the University of Florida stereotactic radiosurgery planning system is the TMR/OAR model [Bov90]. This model calculates dose along the central axis of the beam at the required depth and modifies the central axis dose by multiplying with a measured off-axis ratio. This is acceptable in small beams, as their nearly parallel pencil kernels produce little or no scatter component. This model has been implemented for microcomputers by Suh [Suh90] for both circular and rectangular fields. As presently used, however, this model only calculates the effects of radiation beams produced by circular apertures and cannot model the effects of dynamic conformal collimation with changing, irregular fields.

Two dosimetry methods to predict such effects have been proposed for this work, the convolution method and the negative field method. Each is investigated in turn to determine if its application is appropriate for conformal stereotactic radiosurgery. Also covered in the initial investigation of dosimetry are the effects of two jaw versus four jaw localization, as illustrated in the previous chapter.

### The Convolution Method

The first dosimetry method investigated is that of convolutions using Fourier transforms, based on work by Mohan et al. [Moh87], and Starkschall [Sta88]. The basis of this method is the fact that convolutions are easily computed by taking the Fourier transforms of the functions to be convolved (a complex function of integration). The transformed functions are point multiplied and the product is inverse transformed to arrive at the convolution of the original two functions. This is analogous to adding the logarithms of two numbers one wishes to multiply and taking the antilogarithm of the sum to arrive at the product.

The following discussion of calculating three dimensional dose distributions is taken from Mohan et al. [Moh87]. The basic dose equation is:

$$D(pt) = D_0 \cdot C_m \cdot C_i \quad (5-1)$$

where  $D(pt)$  is the dose in the patient,  $D_0$  is the dose at the same point in a flat, homogeneous, tissue equivalent phantom for an open field of the same size and incident normally on the phantom (obtained from table lookup and interpolation),  $C_m$  is the correction due to beam modifiers, and  $C_i$  is the correction for inhomogeneity and surface irregularities (unity for small beam stereotactic radiosurgery).

$C_m$  may be calculated by:

$$C_m \approx \frac{D_{m,c}}{D_{0,c}} \quad (5-2)$$

where  $D_{m,c}$  and  $D_{0,c}$  are found at the given depth by convolving the relative primary fluence distribution with the profile of the pencil beam distribution at the same depth.

The dose for open or modified fields may then be written as:

$$D_c(x, y, d) = \iint \Phi(a, b) K(x-a, y-b, d) da db \quad (5-3)$$

where  $D_c$  is either  $D_{m,c}$  (modified field) or  $D_{0,c}$  (open field),  $x, y, a, b$  are the lateral distances from the central axis (cm),  $\Phi$  is the relative fluence distribution for the open or modified field, and  $K$  is the two dimensional cross-section profile of the pencil beam at depth  $d$  (the convolution kernel).

Equation 5-3 can be re-written in terms of Fourier transforms as:

$$F\{D_c(x, y, d)\} = F\{\Phi(x, y)\} \cdot F\{K(x, y, d)\} \quad (5-4)$$

where  $F$  signifies taking the two dimensional Fourier transforms of the quantities in braces.

The initial point source fluence can be approximated by a relative fluence of unity at all points inside the open beam and by the collimator transmission at points outside of the open beam. A second point source fluence matrix is

created in which all values of the first have been exponentially attenuated according to the path length of the rays originating from the point source through the beam shaping blocks (approximated by the narrow beam transmission factors of the blocks). To be noted here is the observation that, for a multileaf collimator model, this second matrix results in a quantized representation of leaf position, i.e. each leaf in the model can have a positioning accuracy no smaller than the real space matrix point separation. This fact will dictate the matrix calculation time, which is a function of both matrix size, corresponding to the desired area of spacial coverage, and point spacing, corresponding to the desired accuracy of leaf positioning.

The source size must be included in the model to account for penumbra effects. This is accomplished by assuming a circular disk for the source and determining how much of the source is visible to each point of computation by calculating the area of the source disk inside the projection of the open part of the beam aperture on the plane of the source using the point of computation as the focal point. At isocenter, a source of radius  $r$  has a radius  $r' = \alpha r$  ( $\alpha = (SAD - STD) / STD$  where  $STD$  is the source-to-tray distance, i.e. the location of the block). The source kernel matrix elements in a circular region of  $r'$  at the center of the matrix are set to a constant value representing the source strength or to unity to normalize, and to

zero elsewhere. The source kernel matrix is then convolved with the point source fluence matrices (open and blocked). For the small source size of a linear accelerator, convolution is unnecessary if the source occupies only one source kernel matrix point.

Mohan arrives at the pencil beam kernel by Monte Carlo calculations, however the same endpoint is possible by taking broad beam profiles at several selected depths and deconvolving the x and y beam profiles to develop the kernel [Chu88b]. The pencil beam kernel is convolved with both the open and blocked beam matrices at each selected depth resulting in two three-dimensional dose matrices. The ratios of the corresponding elements of the matrices give a three dimensional matrix of  $C_m$  values which are used in equation 5-1, with interpolation to find doses between selected points.

For initial investigation of this model, a two dimensional fast Fourier transform (FFT) routine by Press et al. [Pre88] was coupled to a driver/timer program. An arbitrary input function was prepared, the timer was started, a forward and reverse transform pair was performed on 16, 32, and 64 square matrices, and the timer was stopped. Test results are shown in table 5-1. The results were extrapolated to find times for realistic matrix sizes in table 5-2.

Using sixteen planes of computation per gantry/table position (4 cm squares spaced at 2.5 cm), and a 512 square

FFT pair (a 5.12 cm square with 0.1 mm spacing, necessary for good resolution of leaf positioning, recalling that in the convolution model the area of spacial coverage is determined by the matrix size and the projected leaf position accuracy is determined by the point spacing), the extrapolated 540 seconds for a single transform pair, i.e. a single gantry/table position and a single arbitrary plane, results in a computation time of 240 hours, excluding interpolations, for a modest 5 arc plan with 100 degree arcs at a 5 degree calculation increment (100 gantry/table positions).

Table 5-1: Experimental 2D FFT Times

Square	Points	Time (sec)
16	512	0.22
32	2048	0.88
64	8192	4.32

Table 5-2: Extrapolated 2D FFT Times

Square	Points	Time (sec)
128	32768	21.6
256	131072	108.0
512	524288	540.0

The FFT matrix size for the completion of a dose volume histogram in a reasonable time of approximately two hours is the 64 square, however this allows leaf positioning to be set to accuracy limits of only 0.625 mm on a 4 cm square



grid, which is more than three times the system average mechanical accuracy.

Further, using the more realistic measure for a rotating model, in which the FFT planes must cover the volume of interest on any projected area, the 4 cm square grid needs a minimum coverage of 6.9 cm on a side. This gives a resolution of 1.08 mm per point for a 64 square FFT matrix, and the same resolution for the leaf settings. This is in contrast to the 0.2 mm average mechanical accuracy of the system, and is even greater than the 0.6 mm pixel resolution of the CT images used for planning [Fri89b] and which would be used for localization.

This analysis shows that the 2D FFT convolution dose model is an inappropriate method for small field rotational dosimetry and work on this model was not continued.

#### The Negative Field Method

The second approach uses the negative field method [Kha70, Kha84] and has the advantage of being a simple modification to a known, verified model. Preliminary investigation of the utility of collimating the beam to the target with this technique was completed using the circular beam model as the standard and modifying the rectangular beam model to calculate blocked irregular fields (both models from Suh [Suh90]). Collimator rotation was not used.

The negative field model derives its name from the calculation technique employed. In this case, a basic

square open field is first calculated with full rotation arcs. Next, each rectangular leaf is treated as if it were an open field (dosimetry is performed over an open field of the dimension each leaf), and the contributions from each are summed over the same arcs. Finally, the sum of the leaf fields is subtracted from the base open field, i.e. the leaf fields act as a "negative" field. This technique includes scatter-air ratios which are part of the measured data at the edges of the leaves, and thus is effectively equivalent to the SAR method and Clarkson integration.

A spherical head phantom of 20 cm in diameter was used with the target at the center. This phantom size and shape is considered appropriate for the head as Pike uses an 18 cm diameter spherical phantom for verification work [Pik90], and the ICRP standard man phantom head is modeled by a 20 x 24 cm right circular cylinder topped by a hemisphere [Ker80]. Four target shapes were modelled: one by a 2 cm diameter sphere at the center of the head phantom; the other three by an ellipsoid (2 cm on the major axis, 1 cm on the minor axes) at the center of the head phantom. The first had the major axis in the AP orientation and the minor axes in the coronal plane, the second had the major axis in the superior/inferior orientation and the minor axes in the axial plane, and the third had the major axis oriented obliquely from anterior/superior to posterior/inferior.

A treatment plan, the same for each target for comparison purposes, was prepared using five non-coplanar parasagittal  $100^\circ$  arcs with  $5^\circ$  incrementation at arbitrary table angles of  $50^\circ$ ,  $70^\circ$ ,  $90^\circ$ ,  $290^\circ$ , and  $310^\circ$ . Minimum margins for each were set at 5 mm. The targets were localized for the conformal cases using 2.5 mm leaves and the plan was run for four jaw conformal collimation, two jaw conformal collimation (with localization in the AB collimator dimension), and for conventional, single isocenter collimation.

Axial, sagittal, and coronal dose distributions through isocenter were computed to visually evaluate the goodness-of-fit of distribution to target. Differential dose volume histograms were computed to quantitatively evaluate the plans. The histograms were further evaluated using the integrated logistic function [Fli89; Fli90b], modified for qualitative comparison (see appendix E).

Additionally, the AP oriented ovoid conformal plan was compared to a two isocenter and a three isocenter plan using the same treatment parameters as above. The plans were compared using dose distributions on the three major planes, by generating three-dimensional dose amplitude plots on the axial and sagittal planes, and also by dose volume histograms and the integrated logistic formula.

Comparing figures 5-1, 5-2, and 5-3 for the case of the spherical target, we note that all the figures display similar isodose patterns. This is confirmed by observing

the dose volume histograms for this case, figure 5-4. In each histogram (total volume, target volume, normal tissue volume) the histograms show similar dosimetry. This demonstrates that the addition of conformal collimation does not degrade system performance already established, and indeed that a spherical target is best fit with a spherical dose distribution. Additionally, the integrated logistic function results for the normal tissue in the calculated volume are also (roughly) similar with values of 0.175 for the 4 jaw localization, 0.407 for the 2 jaw localization, and 0.299 for the circular field for prescribed doses of 1000 cGy to the 70% line for each. Note that in the integrated logistic function comparison, lower numbers are defined as better (although only qualitatively better) and that no evaluation of homogeneity within the target volume is performed.

Figures 5-5, 5-6, and 5-7, the AP oriented ovoid, show great improvement for the 4 jaw conformal collimation versus the 2 jaw or the single isocenter circular, with the dosimetry of the 2 jaw and the circular being fairly similar. This is confirmed by observing the dose volume histogram, figure 5-8, and by evaluating the integrated logistic function. This evaluation gives values of  $<0.001$  for the 4 jaw, 0.360 for the 2 jaw, and 0.391 for the circular.

Figures 5-9, 5-10, and 5-11, for the axial ovoid, show steps of improvement, with the best fit being produced by

the 4 jaw collimation, followed by 2 jaw, and then by circular. This is quantitatively confirmed by the dose volume histogram comparison, figure 5-12, and by the integrated logistic function results:  $<0.001$  for the 4 jaw; 0.231 for the 2 jaw; and 0.428 for the circular.

These results are echoed by figures 5-13, 5-14, and 5-15, for the oblique ovoid. Again, the 4 jaw collimation produces the best results, followed by the 2 jaw, and trailed by the circular. The dose volume histogram in figure 5-16 also shows this. Calculating the integrated logistic function for these volumes gives:  $<0.001$  for the 4 jaw; 0.274 for the 2 jaw; and 0.397 for the circular.

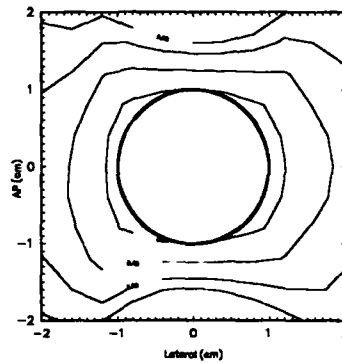
Finally, the AP ovoid is localized with a 4 jaw collimator and the resulting dosimetry is compared with two isocenter and three isocenter treatment plans in figures 5-17, 5-18, and 5-19. This comparison is important in that the common method for producing conformal dosimetry at the present time is by employing multiple isocenters. Observing these figures shows similar conformation in the high isodose regions, with the low isodose lines on the multiple isocenter plots being much more spread out. Also of importance is the observation that the conformally collimated plan has the target enclosed in the 80% isodose line, whereas the multi-isocenter plans reduce coverage to the 70% line. This reduction is common for any multi-isodose plan. Homogeneity of coverage is graphically illustrated in figures 5-20, 5-

21, and 5-22 for the multileaf collimator, the two isocenter, and the three isocenter plans, respectively. The multileaf collimator quite obviously produces a homogeneous dose across the target. The two and three isocenter plans show the characteristic peaks in dose where the edges of the isodose spheres produced by the circular collimators overlap. The dose volume histogram, figure 5-23, also decisively shows the difference, with the target volume dose volume histogram reflecting these peaks and valleys. The integrated logistic function computes values of  $<0.001$  for the conformal collimator, 0.154 for the two isocenter plan, and 0.196 for the three isocenter plan.

#### Conclusion

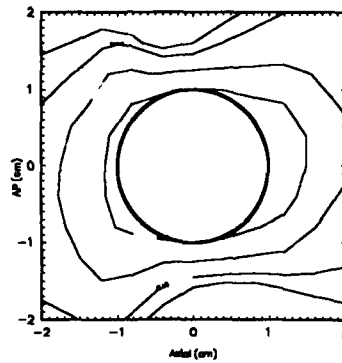
The convolution method has been shown to be inappropriate for use with small field stereotactic radiosurgery because of the tradeoffs between accuracy and time. The negative field method will form the basis of the dosimetry to be further developed in this work. Each of the preceding dosimetry comparisons shows the superiority of conformal collimation, 4 jaw conformal collimation in particular, to single or multiple isocenter treatment plans, and the feasibility of such multileaf collimator planning.

Conformal Collimation, 4 Jaw  
Axial, SPHERE.CT



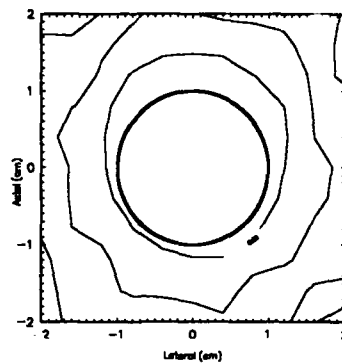
(a)

Conformal Collimation, 4 Jaw  
Sagittal, SPHERE.CT



(b)

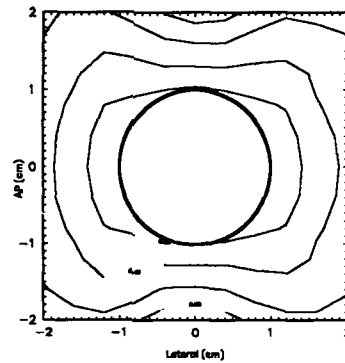
Conformal Collimation, 4 Jaw  
Coronal, SPHERE.CT



(c)

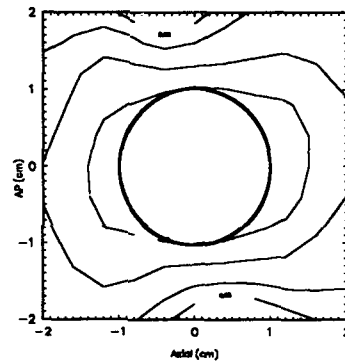
Figure 5-1: Four jaw conformal collimation, sphere  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane; (b) Sagittal plane; (c) Coronal plane

Conformal Collimation, 2 Jaw  
Axial, SPHERE.CT



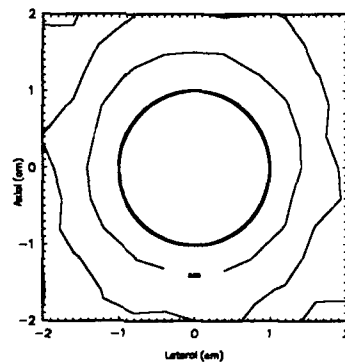
(a)

Conformal Collimation, 2 Jaw  
Sagittal, SPHERE.CT



(b)

Conformal Collimation, 2 Jaw  
Coronal, SPHERE.CT

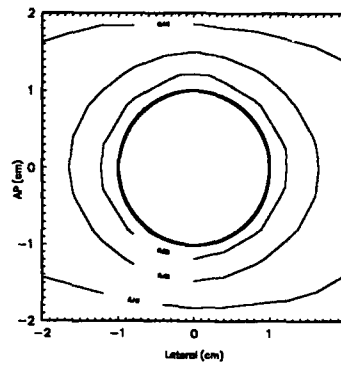


(c)

Figure 5-2: Two jaw conformal collimation, sphere  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane; (b) Sagittal plane; (c) Coronal plane

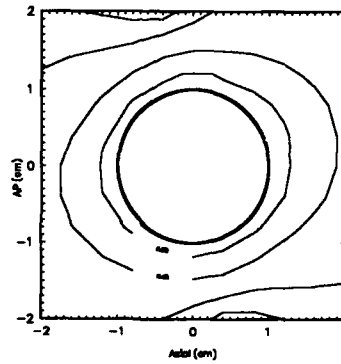


Conventional Collimation, 1 isocenter  
Axial, SPHERE.CT



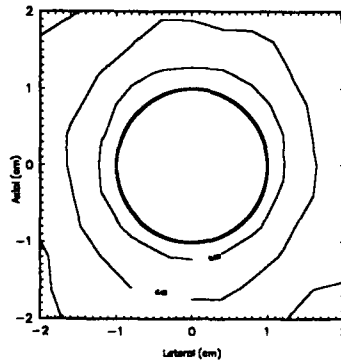
(a)

Conventional Collimation, 1 isocenter  
Sagittal, SPHERE.CT



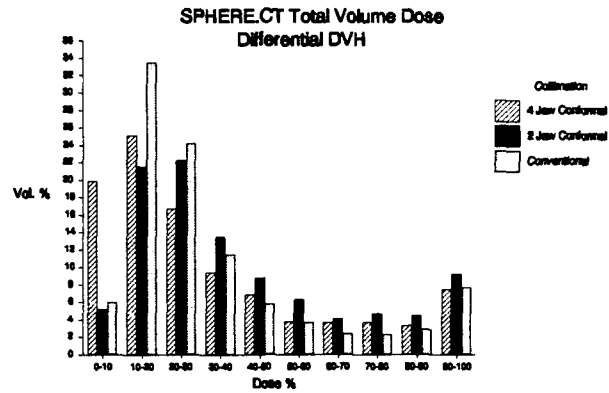
(b)

Conventional Collimation, 1 isocenter  
Coronal, SPHERE.CT

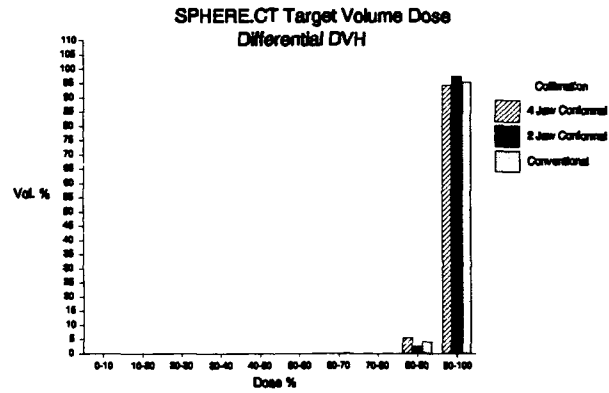


(c)

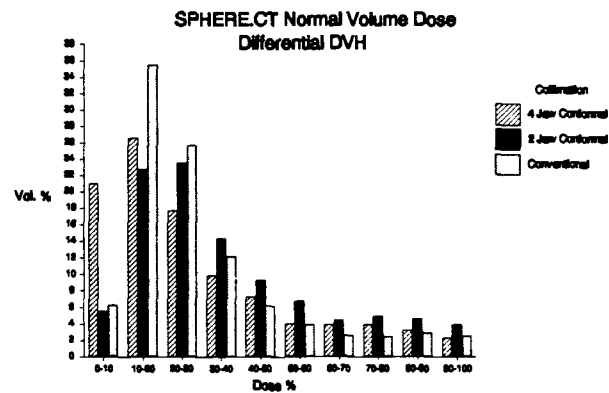
Figure 5-3: One center circular collimation, sphere  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane; (b) Sagittal plane; (c) Coronal plane



(a)



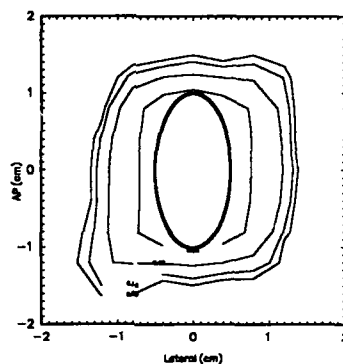
(b)



(c)

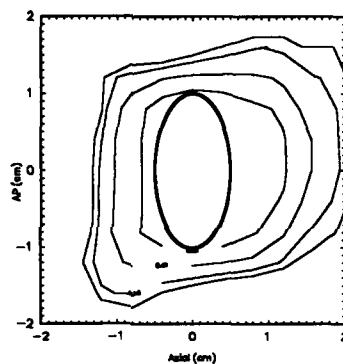
Figure 5-4: Dose volume histograms, sphere  
(a) Total volume; (b) Target volume; (c) Normal volume

Conformal Collimation, 4 Jaw  
Axial, APOVD.CT



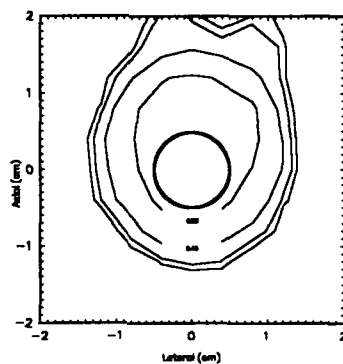
(a)

Conformal Collimation, 4 Jaw  
Sagittal, APOVD.CT



(b)

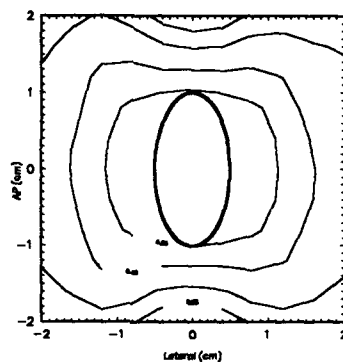
Conformal Collimation, 4 Jaw  
Coronal, APOVD.CT



(c)

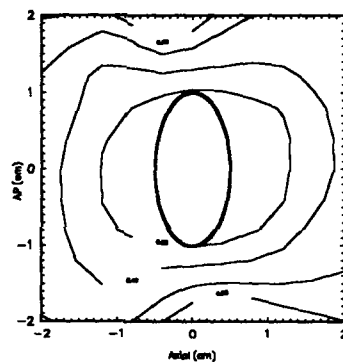
Figure 5-5: Four jaw conformal collimation, AP ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conformal Collimation, 2 Jaw  
Axial, APOVD.CT



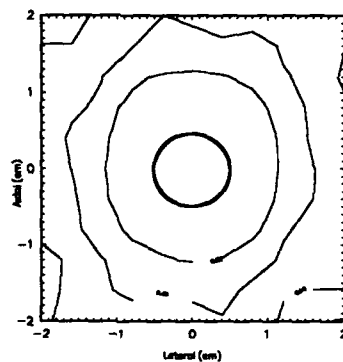
(a)

Conformal Collimation, 2 Jaw  
Sagittal, APOVD.CT



(b)

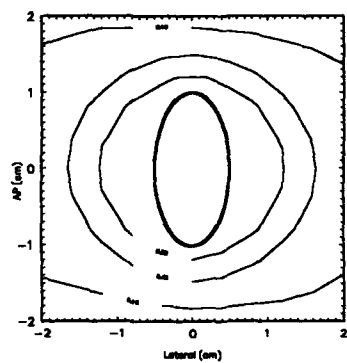
Conformal Collimation, 2 Jaw  
Coronal, APOVD.CT



(c)

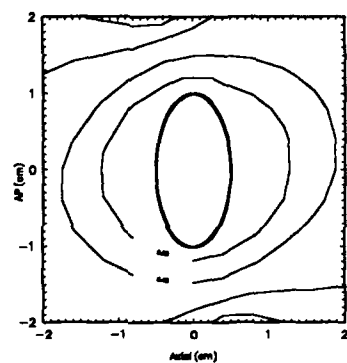
Figure 5-6: Two jaw conformal collimation, AP ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conventional Collimation, 1 Isocenter  
Axial, APOVD.CT



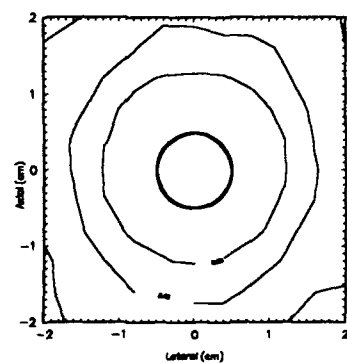
(a)

Conventional Collimation, 1 Isocenter  
Sagittal, APOVD.CT



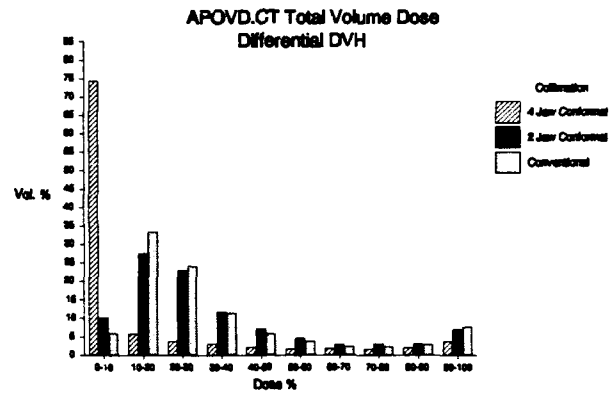
(b)

Conventional Collimation, 1 Isocenter  
Coronal, APOVD.CT

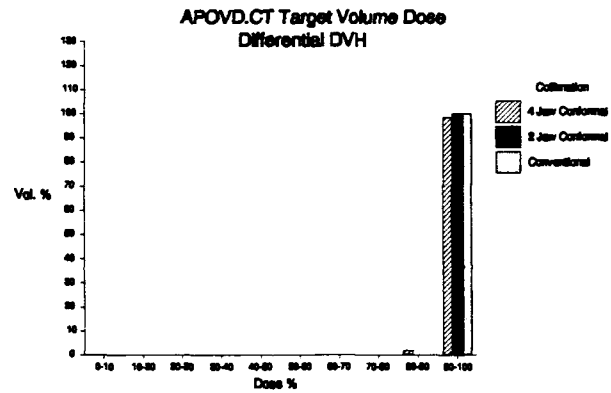


(c)

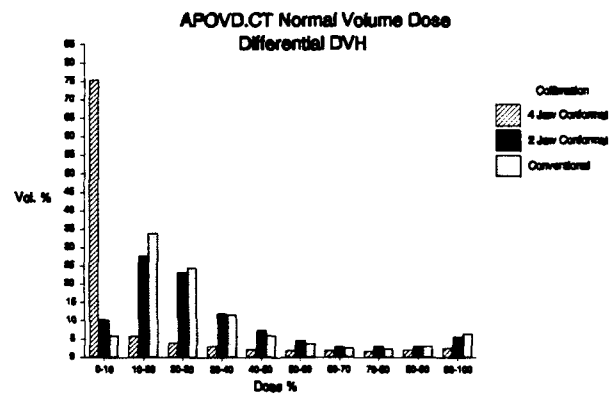
Figure 5-7: One center circular collimation, AP ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane



(a)



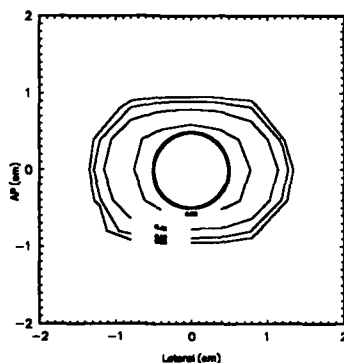
(b)



(c)

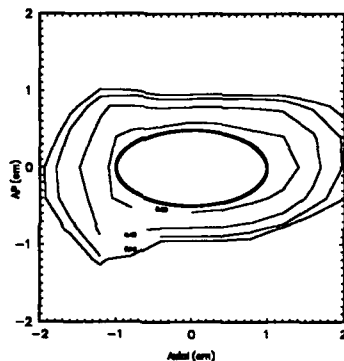
Figure 5-8: Dose volume histograms, AP ovoid  
 (a) Total volume; (b) Target volume; (c) Normal volume

Conformal Collimation, 4 Jaw  
Axial, AXOVD.CT



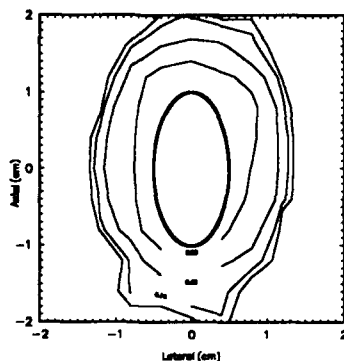
(a)

Conformal Collimation, 4 Jaw  
Sagittal, AXOVD.CT



(b)

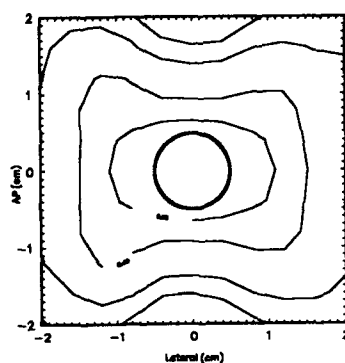
Conformal Collimation, 4 Jaw  
Coronal, AXOVD.CT



(c)

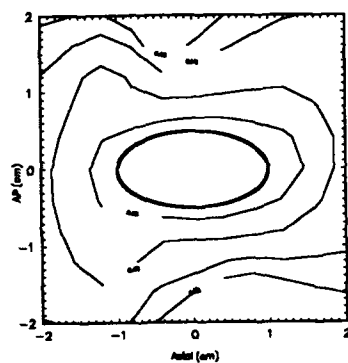
Figure 5-9: Four jaw conformal collimation, axial ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conformal Collimation, 2 Jaw  
Axial, AXOVD.CT



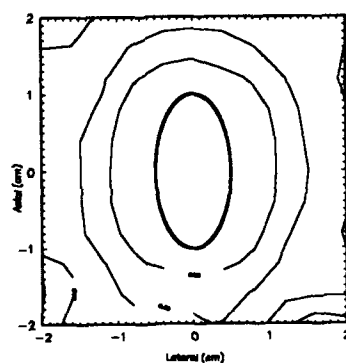
(a)

Conformal Collimation, 2 Jaw  
Sagittal, AXOVD.CT



(b)

Conformal Collimation, 2 Jaw  
Coronal, AXOVD.CT

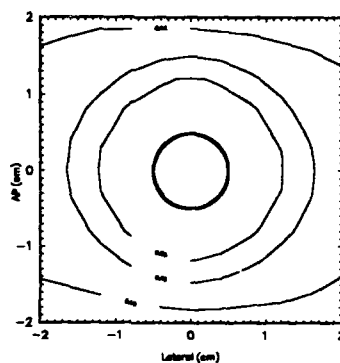


(c)

Figure 5-10: Two jaw conformal collimation, axial ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

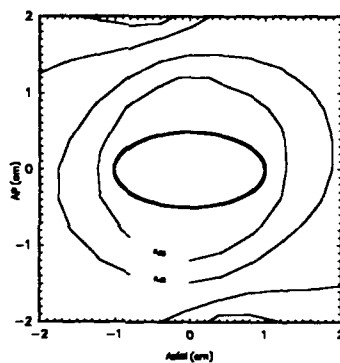


Conventional Collimation, 1 Isocenter  
Axial, AXOVD.CT



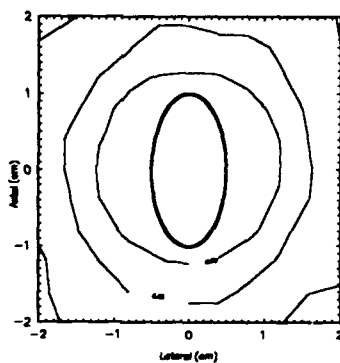
(a)

Conventional Collimation, 1 Isocenter  
Sagittal, AXOVD.CT



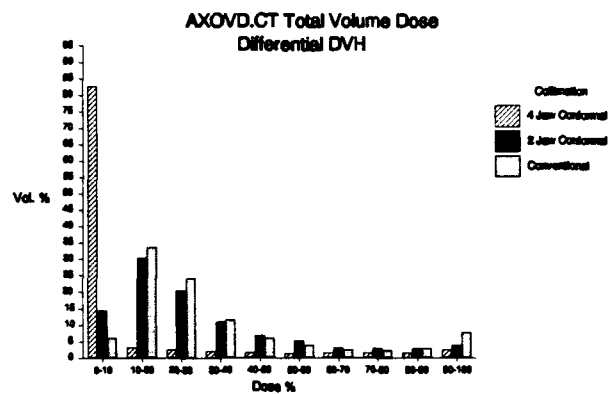
(b)

Conventional Collimation, 1 Isocenter  
Coronal, AXOVD.CT

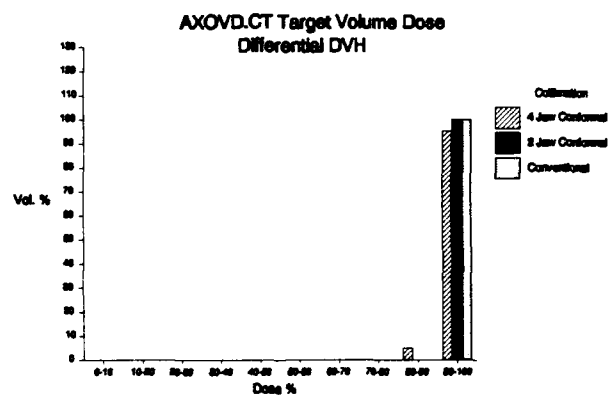


(c)

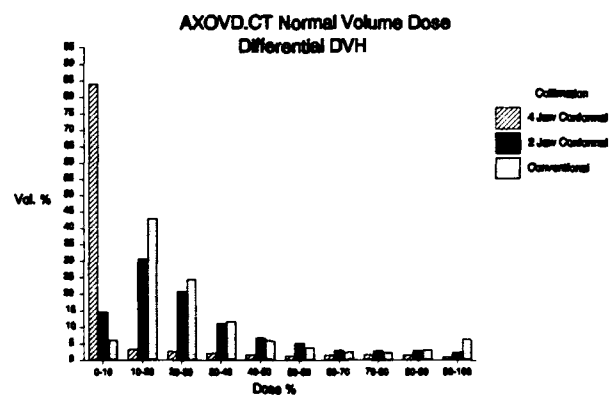
Figure 5-11: One center circular collimation, axial ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane



(a)



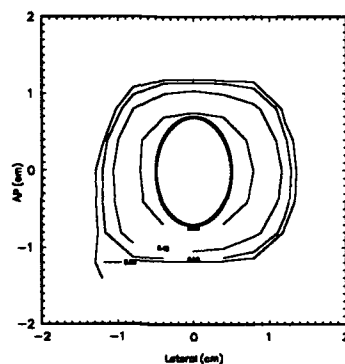
(b)



(c)

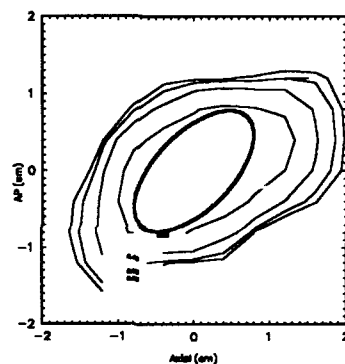
Figure 5-12: Dose volume histograms, axial ovoid  
 (a) Total volume; (b) Target volume; (c) Normal volume

Conformal Collimation, 4 Jaw  
Axial, OBLOVD.CT



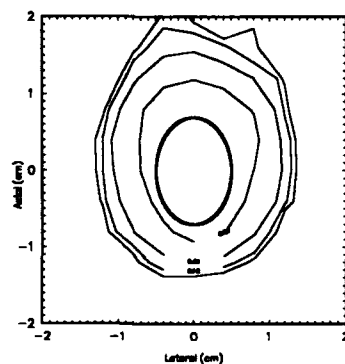
(a)

Conformal Collimation, 4 Jaw  
Sagittal, OBLOVD.CT



(b)

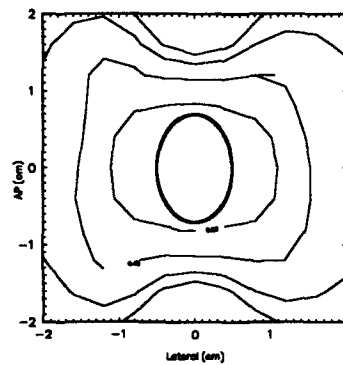
Conformal Collimation, 4 Jaw  
Coronal, OBLOVD.CT



(c)

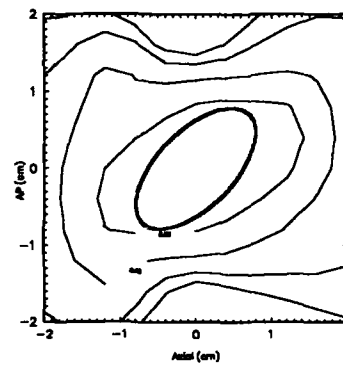
Figure 5-13: Four jaw conformal collimation, oblique ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conformal Collimation, 2 Jaw  
Axial, OBLQVD.CT



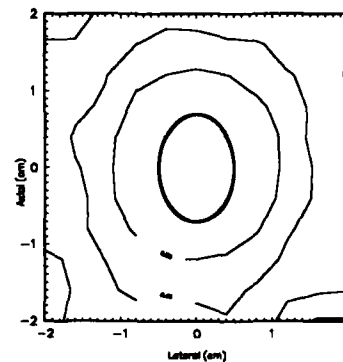
(a)

Conformal Collimation, 2 Jaw  
Sagittal, OBLQVD.CT



(b)

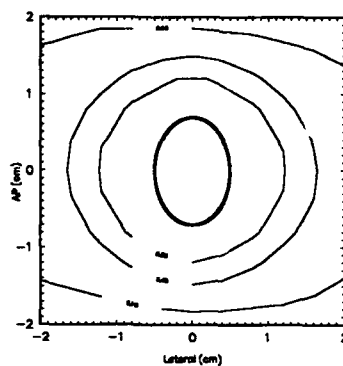
Conformal Collimation, 2 Jaw  
Coronal, OBLQVD.CT



(c)

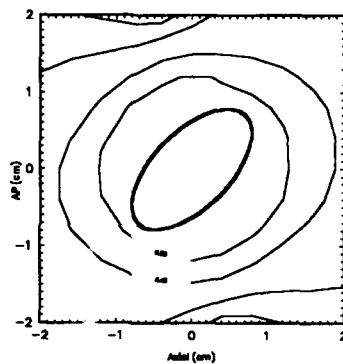
Figure 5-14: Two jaw conformal collimation, oblique ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conventional Collimation, 1 Isocenter  
Axial, OBLOVD.CT



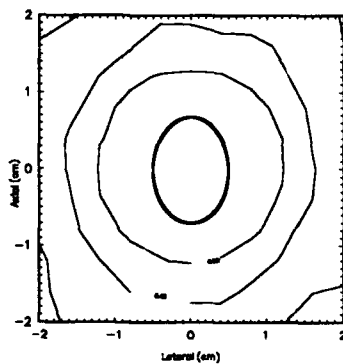
(a)

Conventional Collimation, 1 Isocenter  
Sagittal, OBLOVD.CT



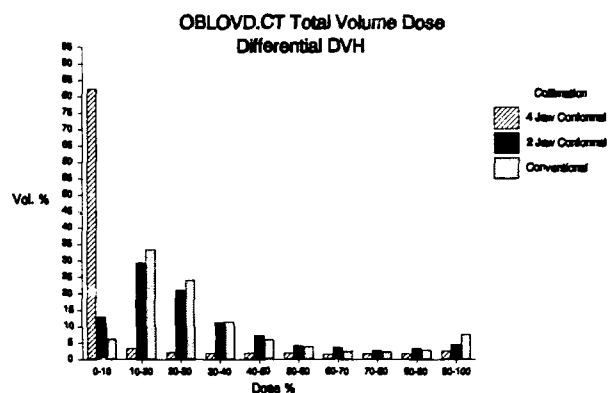
(b)

Conventional Collimation, 1 Isocenter  
Coronal, OBLOVD.CT

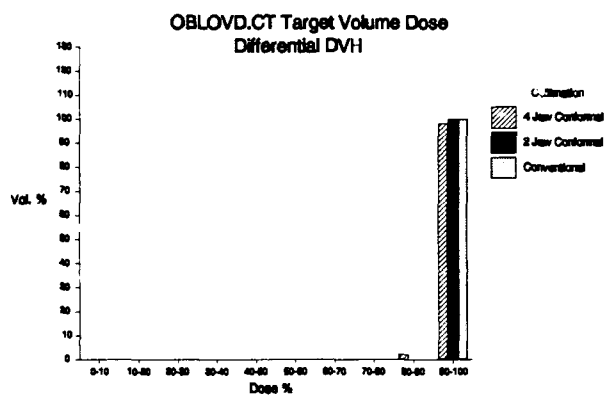


(c)

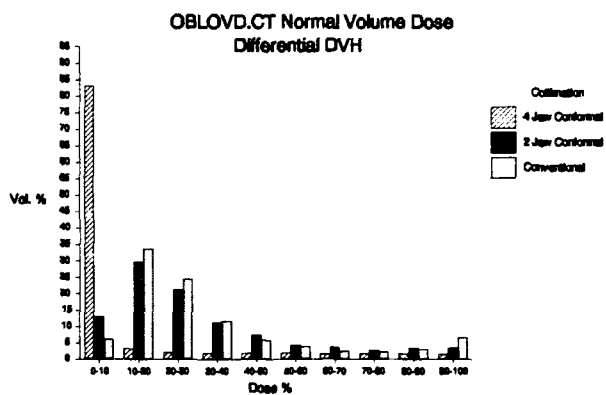
Figure 5-15: One center circular collimation, oblique cvoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane



(a)



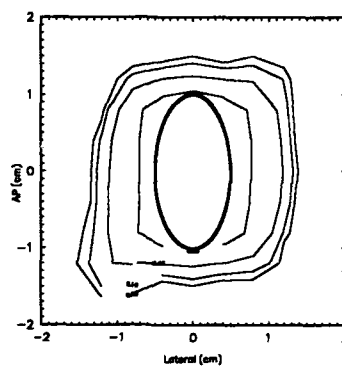
(b)



(c)

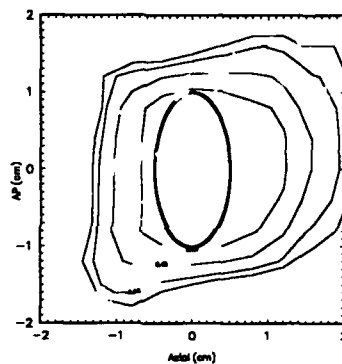
Figure 5-16: Dose volume histograms, oblique ovoid  
(a) Total volume; (b) Target volume; (c) Normal volume

Conformal Collimation, 4 Jaw  
Axial, APOVD.CT



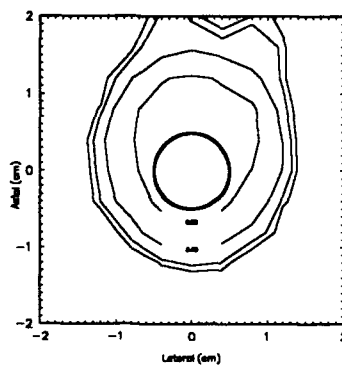
(a)

Conformal Collimation, 4 Jaw  
Sagittal, APOVD.CT



(b)

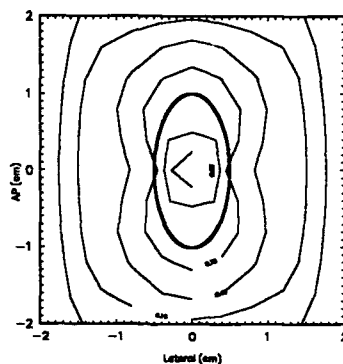
Conformal Collimation, 4 Jaw  
Coronal, APOVD.CT



(c)

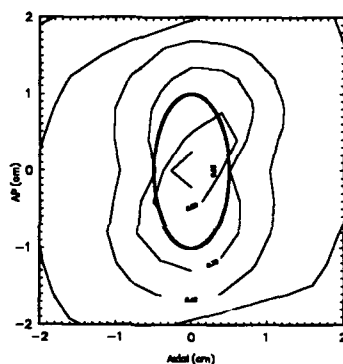
Figure 5-17: Four jaw conformal collimation, AP ovoid  
80, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conventional Collimation, 2 Isocenter  
Axial, APOVD.CT



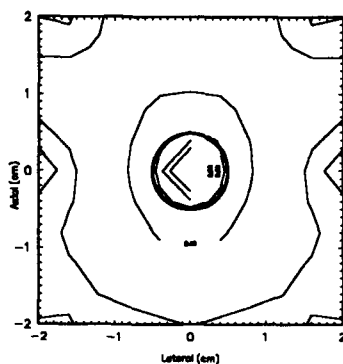
(a)

Conventional Collimation, 2 Isocenter  
Sagittal, APOVD.CT



(b)

Conventional Collimation, 2 Isocenter  
Coronal, APOVD.CT

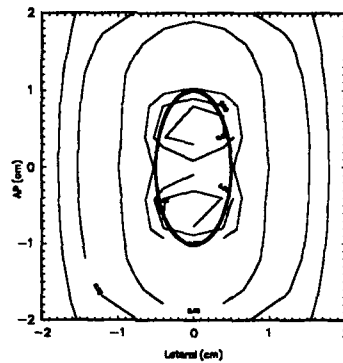


(c)

Figure 5-18: Two center circular collimation, AP ovoid  
90, 80, 70, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

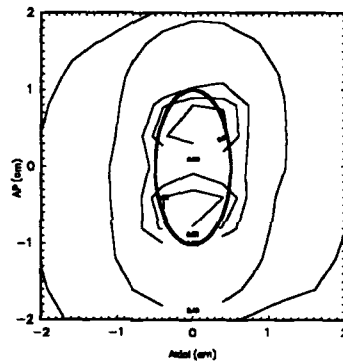


Conventional Collimation, 3 Isocenter  
Axial, APOVD.CT



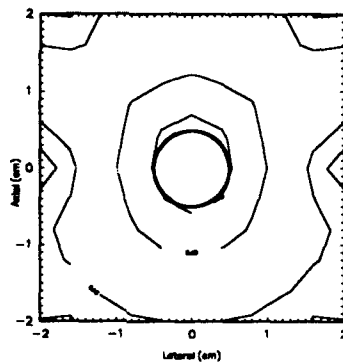
(a)

Conventional Collimation, 3 Isocenter  
Sagittal, APOVD.CT



(b)

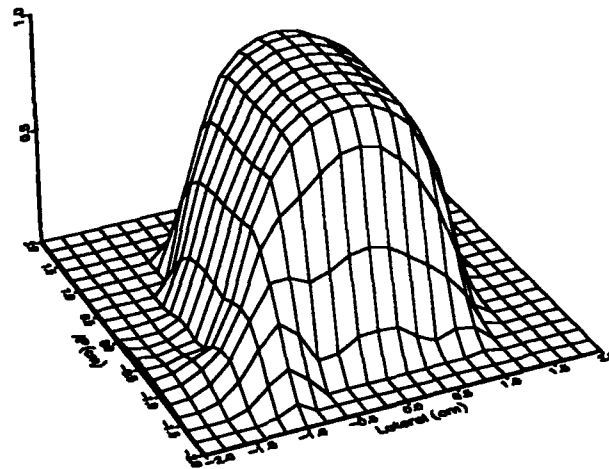
Conventional Collimation, 3 Isocenter  
Coronal, APOVD.CT



(c)

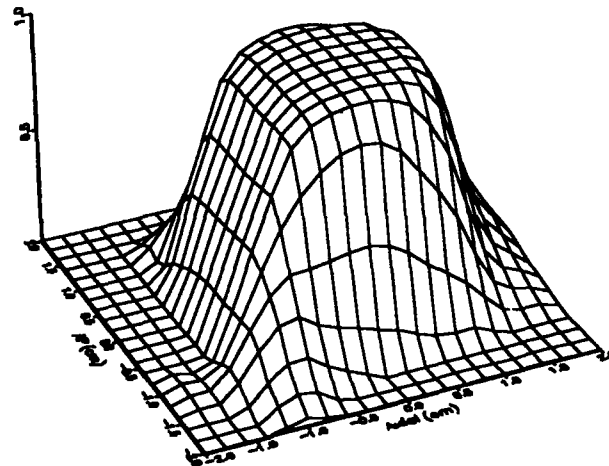
Figure 5-19: Three center circular collimation, AP ovoid  
90, 80, 70, 40, 16, 8% lines normalized to plane maximum  
(a) Axial plane, (b) Sagittal plane, (c) Coronal plane

Conformal Collimation  
Axial, APOVD.CT



(a)

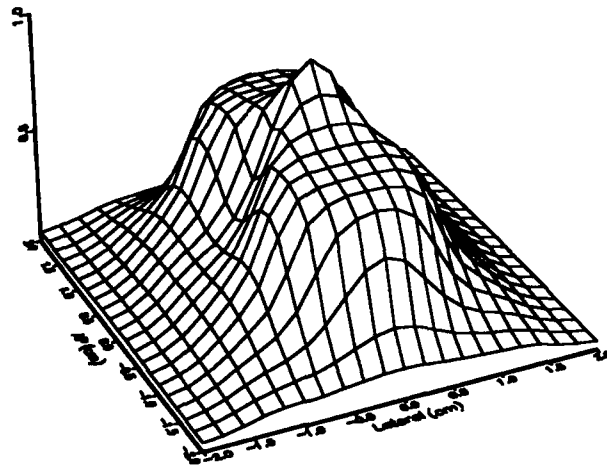
Conformal Collimation  
Sagittal, APOVD.CT



(b)

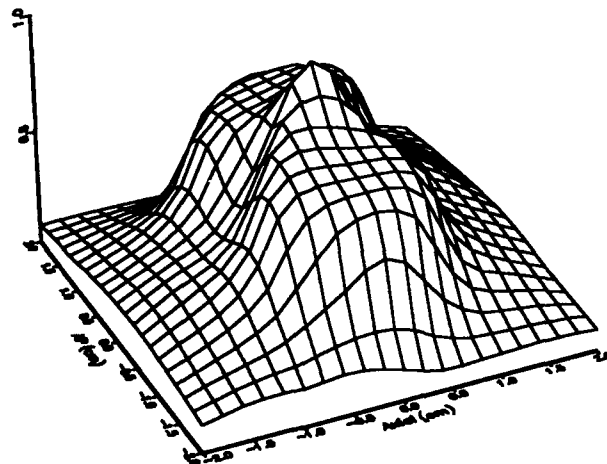
Figure 5-20: Conformal collimation, AP ovoid  
(a) Axial plane; (b) Sagittal plane

Circular Collimation, 2 Isocenter  
Axial, APOVD.CT



(a)

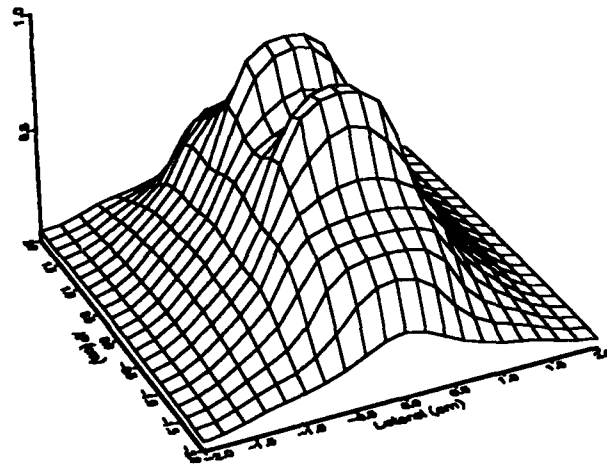
Circular Collimation, 2 Isocenter  
Sagittal, APOVD.CT



(b)

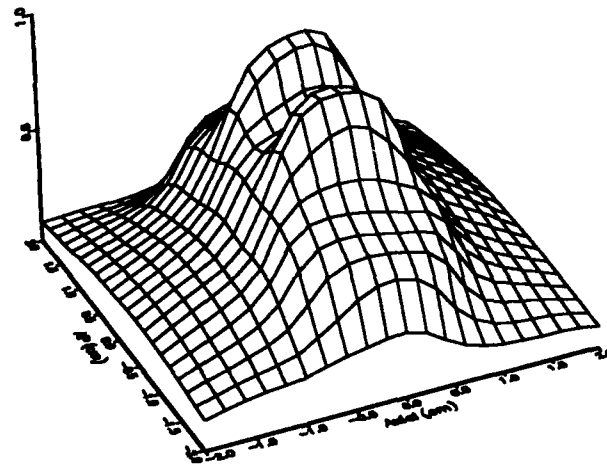
Figure 5-21: Two center circular collimation, AP ovoid  
(a) Axial plane; (b) Sagittal plane

Circular Collimation, 3 Isocenter  
Axial, APOVD.CT



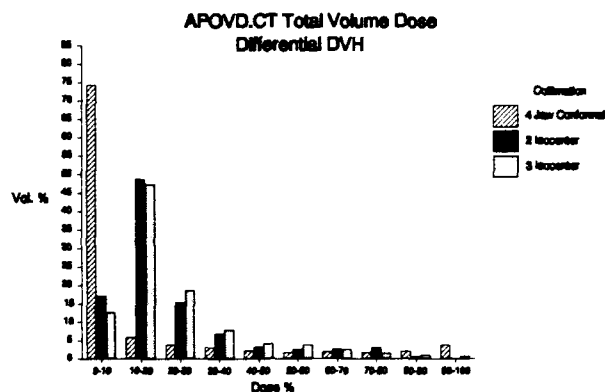
(a)

Circular Collimation, 3 Isocenter  
Sagittal, APOVD.CT

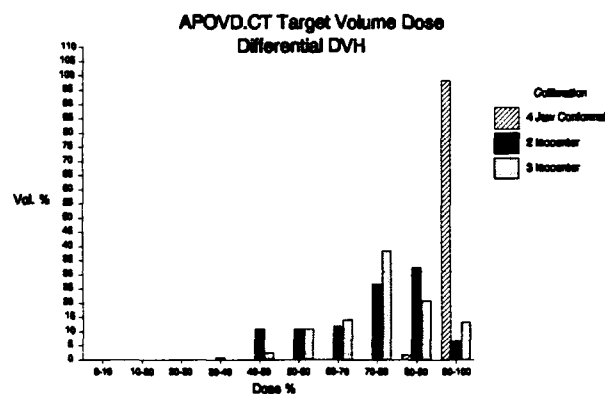


(b)

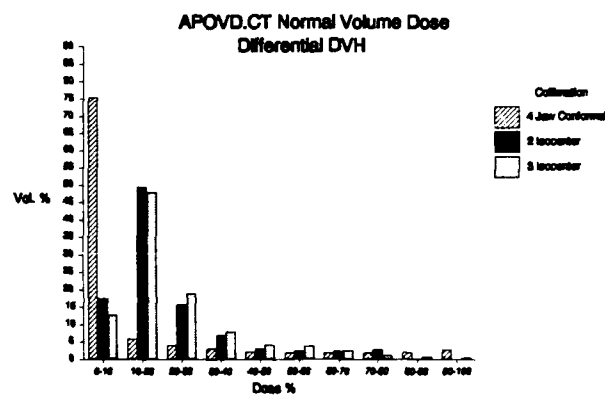
Figure 5-22: Three center circular collimation, AP ovoid  
(a) Axial plane; (b) Sagittal plane



(a)



(b)



(c)

Figure 5-23: Dose volume histograms, AP ovoid  
(a) Total volume; (b) Target volume; (c) Normal volume

## CHAPTER 6

### THE MODIFIED NEGATIVE FIELD METHOD

The negative field method [Kha70; Kha84] used in chapter 5 was modified by the dosimetry model of Chui et al. [Chu86; Chu88a]. The resultant combined system is termed the modified negative field method. This method incorporates measured beam output factors for field size dependence, tissue maximum ratios for depth, off-center ratios for deviation from the beam central axis, and collimator boundary factors for computational accuracy. The following equations which form the basis of this method are from Chui [Chu88a].

The basic dose equation is:

$$D_p = D_m \cdot OF(a_0) \cdot TMR(d, a) \cdot G \cdot OCR(x, y, d) \quad (6-1)$$

where  $D_m$  is the machine dose at a reference point for a reference field (generally 10 x 10 cm),  $OF(a_0)$  is the output factor (field size dependence) at  $d_{max}$  in a phantom for a field with area/perimeter  $a_0$  cm,  $TMR(d, a)$  is the tissue maximum ratio at depth  $d$  on the beam central axis for a field with area/perimeter  $a$  cm,  $G$  is the inverse square factor ( $[SAD/(SSD+d)]^2$ ), and  $OCR(x, y, d)$  is the off center ratio at  $P(x, y, d)$ , the ratio of the dose at point  $P(x, y)$ , where  $x$  and  $y$  are the orthogonal off axis coordinates, to

the dose at a point on the collimator central axis at the same depth  $d$ .

The OCR is found by multiplying the primary OCR (POCR) for a quasi-infinite (40 x 40 cm square) beam and the boundary factors of all four field sides:

$$OCR = POCR(r, d) \cdot \prod_{i=1}^4 BF(s_i, d, w, h) \quad (6-2)$$

where  $r$  is the radial distance from the beam central axis (cm),  $d$  is the depth (cm),  $s_i$  is the distance (cm) from field edge  $i$  ( $1 \leq i \leq 4$ ), and  $w, h$  are field width and height (cm).

Table 6-1: Measurement equipment

Type	Company	Model	Serial	Active Vol/Area
Ion chamber	PTW	N23323	2914	0.1 cc
Electrometer	Keithley	35614	38186	NA
Photon diode	Nuclear Associates	30-490-8	07463	2 mm dia circle
Electron diode	Nuclear Associates	30-495-8	11070	2 mm dia circle
Electrometer	Keithley	602	65505A	NA

Basic beam data measurements were carried out on a Philips SL75-5 linear accelerator located at the Shands Cancer Center at the University of Florida with an x-ray energy of 6 MV. The equipment used is listed in table 6-1. The ion chamber was used as a standard and the diodes were checked against the ion chamber readings. The electron

diode was found to track the ion chamber to within 2% for readings at the surface to a depth of 2 cm. The photon diode was found to respond appropriately, to within 1.5%, at depths of 1.5 cm and deeper.

#### Output Factor

The output factor was measured with the electron diode at the depth of maximum dose,  $d_{\max}$  (1.5 cm for 6 MV x-rays), in a solid water phantom for square fields of 1, 2, 3, 4, 5, and 10 cm (area/perimeter ratios of 0.25, 0.5, 0.75, 1.0, 1.25, and 2.5 respectively) at isocenter. The four smaller field sizes were produced by forming machined, 5 cm thick lead bricks into scaled, non-divergent squares at 70 cm from the source. These bricks have a measured transmission factor of 7.8%. The two larger sizes used the treatment machine secondary collimators.

Diode readings were normalized to the readings produced by the 10 x 10 cm field and were extrapolated to zero to account for small leaf widths. Figure 6-1 shows the normalized output factor data plotted against field size normalized to the field area/perimeter. The data was fit to the equation:

$$OF = 1 - 0.4371 e^{-2.627 AP} \quad (6-3)$$

where AP is the area/perimeter normalized field size. The sum of the square errors (SSE) for this fit is  $6.72 \times 10^{-4}$ .



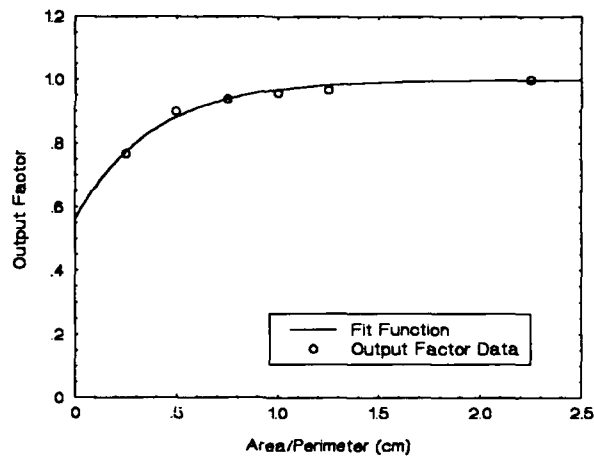


Figure 6-1: Output factor

#### Tissue Maximum Ratio

The tissue maximum ratio (TMR) was measured in solid water with the electron diode in the buildup region and with the photon diode in the equilibrium region. Measurements were made for square field sizes of 1, 2, 3, and 4 cm, again formed by the machined lead bricks. Measurements for each field size were normalized to  $d_{\max}$ . Figure 6-2 shows the results of this series of measurements with the TMR plotted against measurement depth for an average of the readings in the buildup region and for each of the field sizes in the equilibrium region. Results were fit to a third degree polynomial in the buildup region ( $0 \leq \text{depth} \leq 2$  cm) and to declining exponentials in the equilibrium region ( $\text{depth} > 2$  cm).

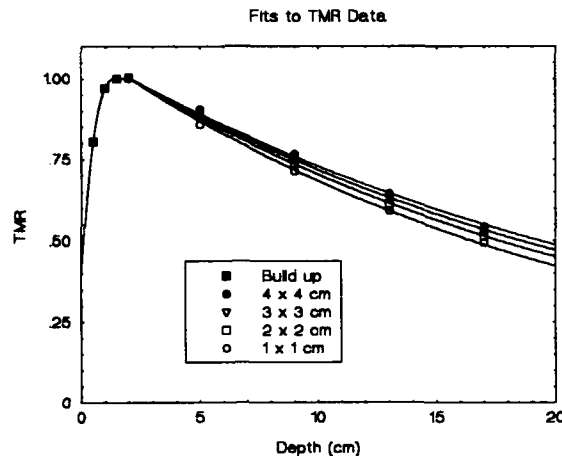


Figure 6-2: Tissue-maximum ratio

The following equations were found:

$$TMR(Bu) = 0.400 + 1.120d - 0.689d^2 + 0.140d^3 \quad (6-4)$$

$$TMR(4 \times 4) = 1.0038 e^{-4.000 \cdot 10^{-2} (d-2)} \quad (6-5)$$

$$TMR(3 \times 3) = 1.0038 e^{-4.193 \cdot 10^{-2} (d-2)} \quad (6-6)$$

$$TMR(2 \times 2) = 1.0038 e^{-4.436 \cdot 10^{-2} (d-2)} \quad (6-7)$$

$$TMR(1 \times 1) = 1.0038 e^{-4.796 \cdot 10^{-2} (d-2)} \quad (6-8)$$

where  $d$  is the depth in cm in the phantom. The SSE's for the TMR equations are  $6.97 \times 10^{-6}$ ,  $3.06 \times 10^{-4}$ ,  $1.91 \times 10^{-4}$ ,  $4.67 \times 10^{-5}$ , and  $2.01 \times 10^{-4}$ , respectively. The constant exponents in equations 6-5 through 6-8 were further fit to account for field size (expressed as area/perimeter). This fit is shown in figure 6-3.

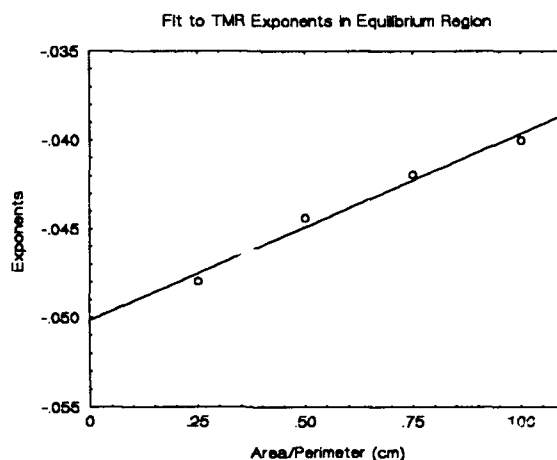


Figure 6-3: Fit to TMR exponents

This fit is linear with the fit equation:

$$\text{Exponent Fit} = -0.0501 + 0.01052 AP \quad (6-9)$$

where AP is the field size expressed as area/perimeter (cm). The SSE for this fit is  $7.20 \times 10^{-7}$ . The exponent fit, equation 6-9, was substituted into equations 6-5 through 6-8 to arrive at a single equation in field size and depth for the equilibrium region:

$$TMR(Eq) = 1.0038 e^{[(-0.0501 + 0.01052 AP)(d-2)]} \quad (6-10)$$

#### Primary Off Center Ratio

The primary off center ratio (POCR) was measured with the electron diode in the buildup region and with the ionization chamber in the equilibrium region using a 40 x 40 (quasi-infinite) field, along both field diagonals and along both axes for radial off axis distances of 0, 1, 2, 3, 4, 5,

6, and 7 cm at depths of 0.5, 1, 1.5, 5, 10, and 15 cm. Readings at each off axis distance were averaged and were normalized to the central axis. Significant differences were found between the measurements in the build up region when compared to the equilibrium region, and the two cases were treated separately in the data analysis.

Figure 6-4 illustrates the measurement of the POCR in the build up region with a plot of off axis ratio vs. distance off axis at two depths. This data was fit to two equations:

$$POCR(0.5) = e^{-2.926 \cdot 10^{-2} OAD} \quad (6-11)$$

$$POCR(1.0) = e^{-6.778 \cdot 10^{-3} OAD} \quad (6-12)$$

where OAD is the off axis distance in cm. The SSE to these equations is  $9.30 \times 10^{-4}$  and  $1.14 \times 10^{-4}$ , respectively.

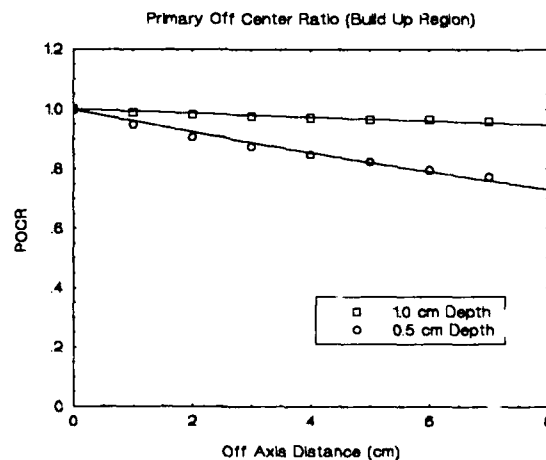


Figure 6-4: Primary off center ratio, build up region

Figure 6-5 shows the POCR, plotted as above, at four depths in the equilibrium region. The equations fitting this data are:

$$POCR(1.5 \text{ cm}) = 2.0 - e^{-5.081 \cdot 10^{-3} OAD} \quad (6-13)$$

$$POCR(5 \text{ cm}) = 2.0 - e^{-4.353 \cdot 10^{-3} OAD} \quad (6-14)$$

$$POCR(10 \text{ cm}) = 2.0 - e^{-1.721 \cdot 10^{-3} OAD} \quad (6-15)$$

$$POCR(15 \text{ cm}) = 2.0 - e^{-1.404 \cdot 10^{-3} OAD} \quad (6-16)$$

where OAD is, again, the distance off of the central axis in cm. The SSE's for these equations are  $1.76 \times 10^{-5}$ ,  $2.61 \times 10^{-5}$ ,  $3.17 \times 10^{-6}$ , and  $4.43 \times 10^{-6}$ , respectively. These equations are fit to off axis distances of 6 cm and less. For distances greater than 6 cm, the value calculated at 6 cm is used as a constant.

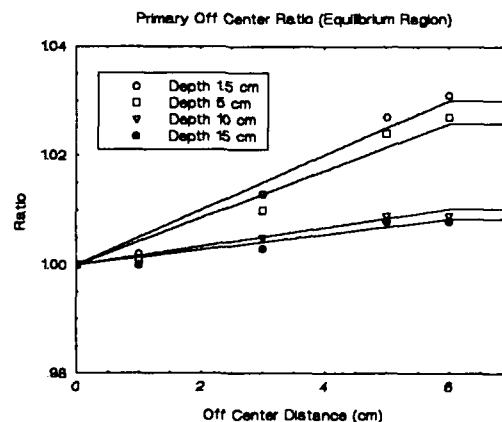


Figure 6-5: Primary off center ratio, equilibrium

Each region's depth data was regressed separately to arrive at two equations for the POCR, one for the build up region and one for the equilibrium region. This process is shown in figures 6-6 and 6-7.

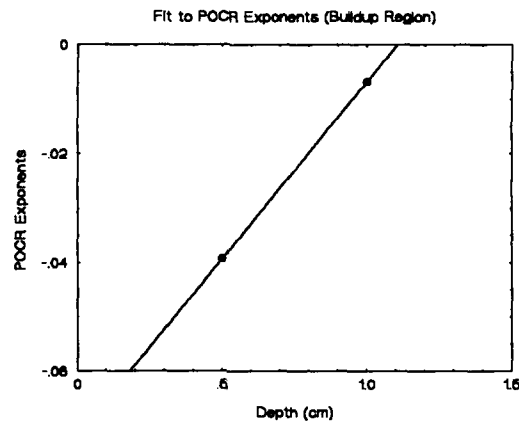


Figure 6-6: Fit to POCR exponents, build up

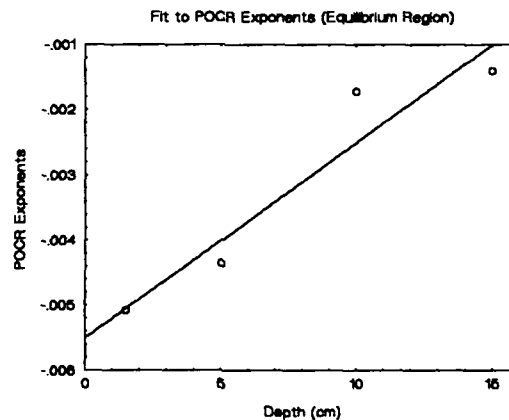


Figure 6-7: Fit to POCR exponents, equilibrium

The fit to the buildup region exponents is:

$$\text{Exponent Fit} = -0.0717 + 0.0650 d \quad (6-17)$$

The fit to the equilibrium region exponents is:

$$\text{Exponent Fit} = -0.00550 + 3.000 \cdot 10^{-4} d \quad (6-18)$$

where  $d$  is the depth in cm in the phantom. The SSE here is  $7.81 \times 10^{-4}$ .

These two exponent fit equations may be substituted, appropriately, in equations 6-11 through 6-16 to arrive at the final POCR equations, separated by region and depth:

$$\text{POCR}(Bu) = e^{[(-0.0717 + 0.0650 d) \text{ OAD}]} \quad (6-19)$$

$$\text{POCR}(Eq) = 2 - e^{[(-0.00550 + 3.000 \cdot 10^{-4} d) \text{ OAD}]} \quad (\text{OAD} \leq 6) \quad (6-20)$$

$$\text{POCR}(Eq) = 2 - e^{[(-0.00550 + 3.000 \cdot 10^{-4} \times d) \times 6.0]} \quad (\text{OAD} > 6) \quad (6-21)$$

### Boundary Factor

The boundary factor accounts for the effects of beam blocking by collimators. It is composed of a combination of geometric penumbra and collimator scatter. There would be no collimator boundary factor (i.e. there would be a step function at the block edge) if the source of the transmitted radiation were infinitesimally small and if there were no collimator scatter. The machined lead blocks previously described were also used for this measurement. Beam cross plots for field sizes 1 x 1, 2 x 2, 3 x 3, and 4 x 4 cm, were gathered at 5 cm depth in solid water with the photon diode. Radiographic films were also taken for each field

size, with appropriate calibration films included. These films were analyzed with a MacBeth film densitometer and optical density was converted to dose by cubic spline fitting to the calibration film data. The beam cross plot data was divided by the POCR at each radial measurement position, as described by Chui [Chu86]. Data was then normalized to the central axis.

Both the diode and the film data were fit to the modified Cunningham model [Phi91]. The modified Cunningham model specifies two paired equations to describe the upper and lower shoulders of the sigmoidally shaped curve produced the collimation of the radiation beam. Both shoulders were similar and the fit exponents were averaged, as shown by the fit lines in the plots. These fits are shown in figures 6-8 through 6-11 for 1 cm through 4 cm square apertures, respectively.

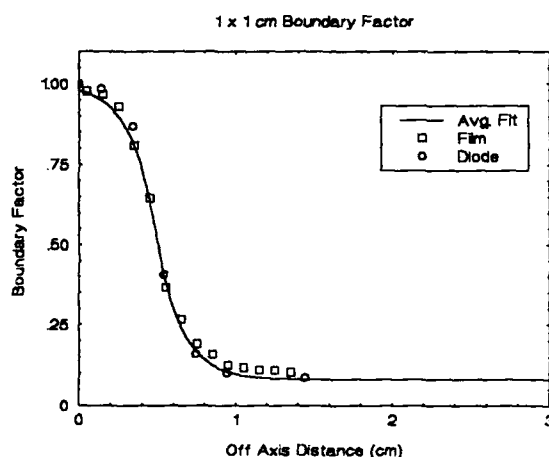


Figure 6-8: Boundary factor, 1 cm square aperture



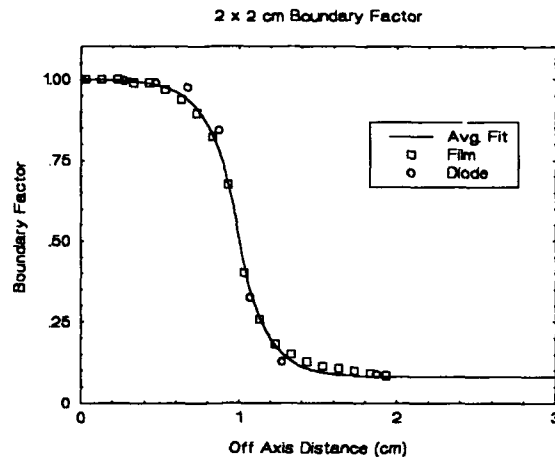


Figure 6-9: Boundary factor, 2 cm square aperture

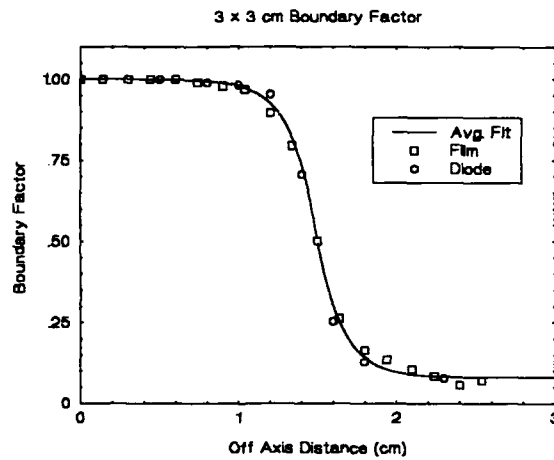


Figure 6-10: Boundary factor, 3 cm square aperture

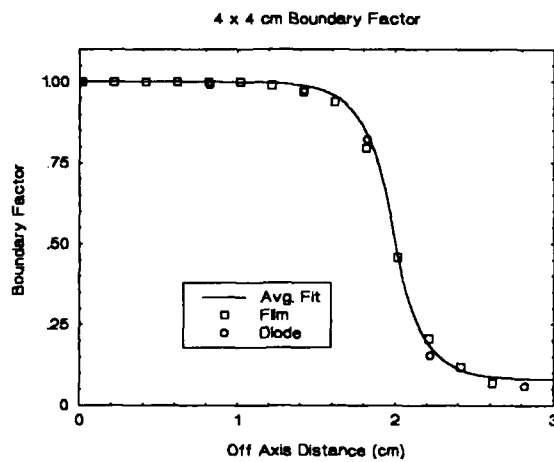


Figure 6-11: Boundary factor, 4 cm square aperture

The equations specified in the model are:

$$BF(upper) = 1 - 0.5 e^{\left[ \alpha_1 \left( \frac{r_0 - r}{p} \right) \right]} \quad (6-22)$$

for  $r \leq r_0$  (the upper shoulder) and

$$BF(lower) = t + (0.5 - t) e^{\left[ \alpha_2 \left( \frac{r_0 - r}{p} \right) \right]} \quad (6-23)$$

for  $r > r_0$  (the lower shoulder) where  $r_0$  is the distance of the collimator edge from the collimator center (cm),  $r$  is the distance of the calculation point from the collimator center (cm),  $t$  is the collimator transmission factor, and  $p$  is the beam penumbra calculated by:

$$p = SS \left( \frac{STD - SCD}{SCD} \right) \quad (6-24)$$

where  $SS$  is the source size (assumed 2 mm for the Philips SL75-5 [Suh90]),  $STD$  is the source to target distance (cm), and  $SCD$  is the source to collimator distance (cm).

The upper shoulder  $\alpha_1$  exponents are similar ( $-0.516 \leq \alpha_1 \leq -0.701$ ) and average to  $-0.6373$ . The lower shoulder  $\alpha_2$  exponents are also similar ( $0.592 \leq \alpha_2 \leq 0.658$ ) and average to  $0.6356$ . These values were used in the final boundary factor equations.

#### The Dosimetry Calculation Process

Equations 6-1 and 6-2 define the dose computation for any point in the volume of interest. The dosimetry programs developed in this work (LFDOSE.C for planar dose computa-

tions in appendix C and LFDVH.C for dose volume histograms in appendix D) are based on the full rotational dosimetry models described by Suh [Suh90] for microcomputers. These models were substantially revised for the computation of dosimetry from dynamic irregular field irradiation and employ the target localization results from chapter 4, recalling that the target localization result is a data series, each datum of which is the position of the leading edge of a rectangular beam block, or a leaf, in the open field.

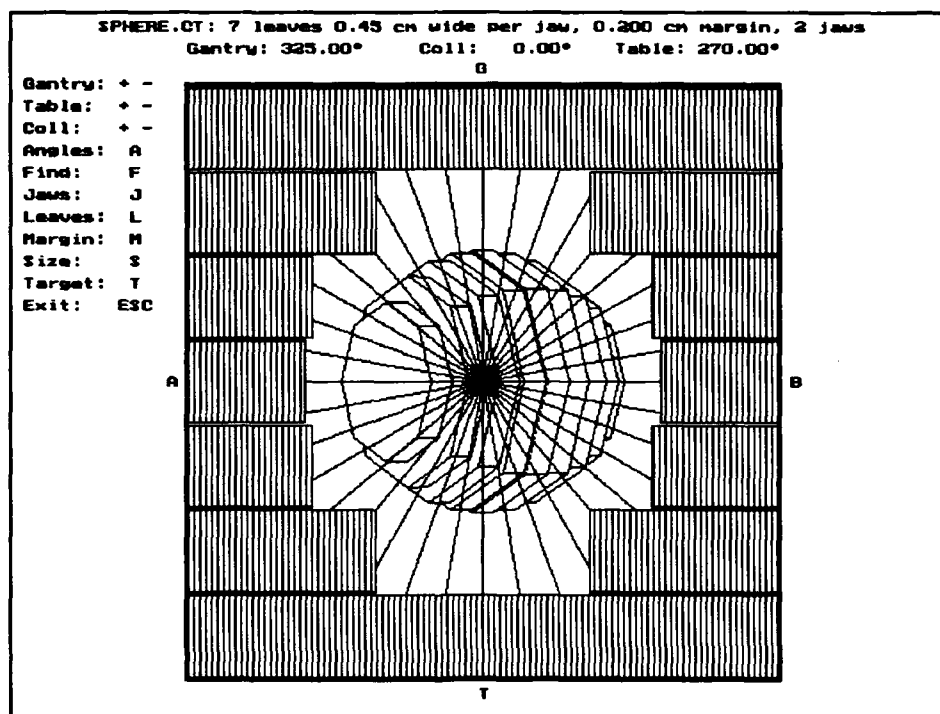


Figure 6-12: Irregular field size determination

Immediately succeeding localization at each gantry/table position the irregular field produced by the process is graphically measured, from the center of the

field to the field edge in  $10^\circ$  increments, to produce an average radius in the manner of Clarkson integration (figure 6-12). The equivalent area/perimeter,  $AP$ , is found from the equation:

$$AP = \frac{\sqrt{\pi}}{4} \times r \quad 6-25$$

where  $r$  is the average radius from the process above [Kha84, page 165]. This is the  $AP$  variable, scaled for distance, that is used in the output factor and tissue-maximum ratio equations.

The algorithm relies on superposition of many discrete fields produced by multiple gantry positions over each arc to simulate a full rotational dosimetry. At each position two matrices are calculated. The first is that produced by a square open field. The second is a summation of fields that would be produced if a rectangle with the dimensions of each discrete leaf in the open field were itself an open field. This second matrix is termed the negative field. As its name implies, after each matrix is calculated, the negative field matrix is subtracted from the open field matrix, resulting in dosimetry from an irregular field that is defined by the positions of the edges of each leaf in the open field. This process follows the equation:

$$D_p = D_o - \sum_{i=1}^{n_l} D_{n_i} \quad (6-26)$$

where  $D_p$  is the dose to the point of interest,  $D_o$  is the dose to the point of interest illuminated by an open field, size defined by the leaf width and the number of leaves used,  $n_l$  is the number of leaves used, and  $D_{n_i}$  is the dose to the point of interest illuminated by a "negative" field, size defined by the leaf width and the distance the leaf intrudes into the open field.

Each dose is calculated using equation 6-1, and uses a collimator transmission factor of 5% to simulate 4 cm thick tungsten leaves. This equation suffices to calculate the point dose at one discrete table/gantry increment.

To begin the dosimetry, matrices for each of the cumulative open fields and negative fields are allocated and zeroed. These matrices are two dimensional in the case of planar dosimetry or three dimensional for dose volume histograms. The table position is set and the gantry is positioned to the bottom of the first arc. The open and negative field matrices are calculated and added to their respective cumulative counterparts. The gantry is then rotated through each arc defined by each table position and open and negative fields are calculated and summed at each. At the conclusion of the rotation plan, the cumulative negative field matrix is subtracted from the cumulative open field matrix resulting in a dosimetry matrix produced by a simu-

lated rotating dynamic irregular field. This matrix is written to a data file suitable for import into a commercial graphics program. The planar program output produces data formatted for a contour plot, the dose volume histogram program output produces bar chart data.

#### Dosimetry Results

The results of the above process are shown in figures 6-13 through 6-18. These results are for, in turn, an AP oriented ovoid, a sphere, an obliquely oriented ovoid, a laterally oriented ovoid, an axially oriented ovoid, and a doubly oblique ovoid. The targets are as described in chapter 4.

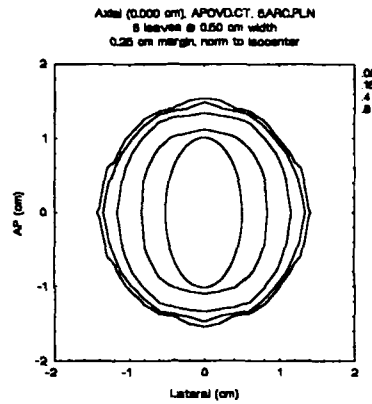
Table 6-2: Standard five arc treatment plan

Arc #	Table Angle	Gantry Start	Gantry Stop	Arc Weight
1	10°	30°	130°	1
2	50°	30°	130°	1
3	350°	230°	330°	1
4	310°	230°	330°	1
5	270°	230°	330°	1

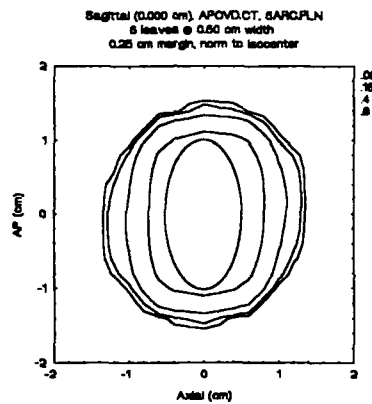
Each target was localized with a 24 element multileaf collimator (6 leaves, 5 mm wide, on each of four sides, producing an open field of 3 cm square at the collimator position) with a margin of 2.5 mm at closest approach. A standard five arc treatment plan, with equally spaced coro-

nal coverage approaching the optimum of  $2\pi$ , was employed (table 6-2). The gantry was rotated in increments of  $5^\circ$ .

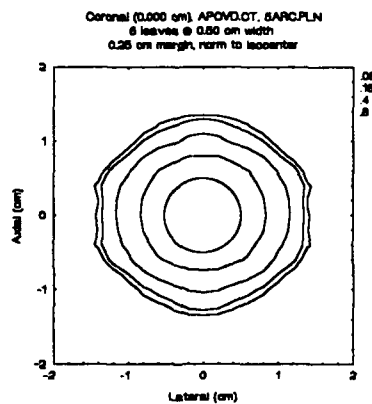
Isodose plots on each of the three principal planes, axial, sagittal and coronal, were produced for each target except the double oblique ovoid. This last was characterized by axial slices at 2 mm intervals over the extent of the target. Each target is fit by the 80% line on each plane and the low isodose lines have minimal spreading. These criteria are perhaps the most important in stereotactic radiosurgery and are well satisfied with this method. Dosimetry verification and treatment plan variability will be covered in subsequent chapters.



(a)



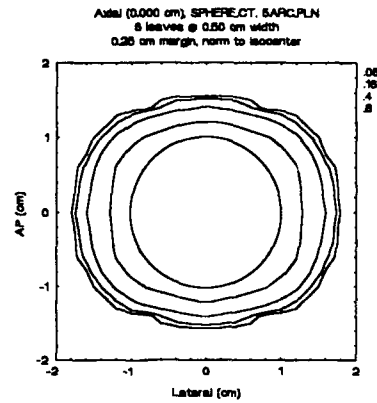
(b)



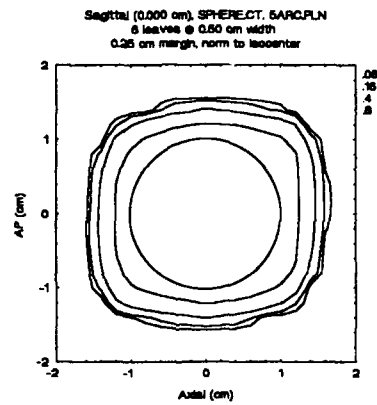
(c)

Figure 6-13: AP oriented ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

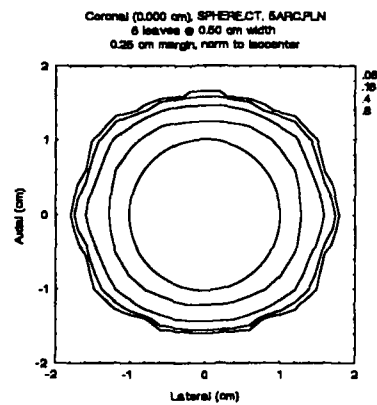




(a)

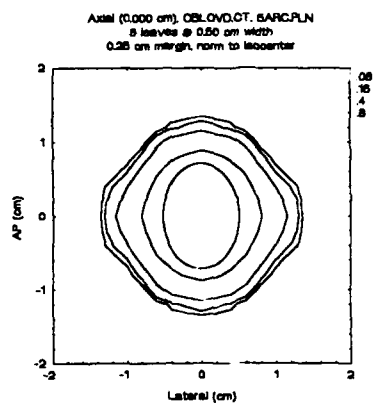


(b)

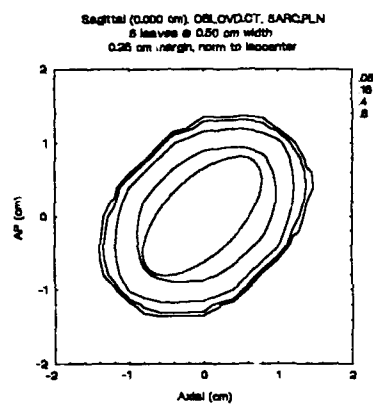


(c)

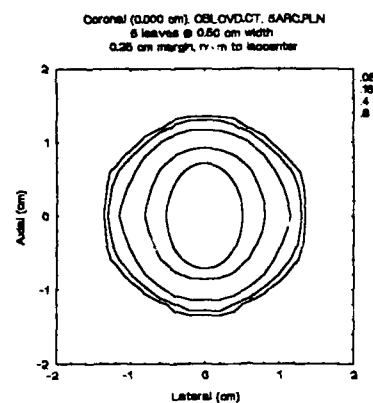
Figure 6-14: Sphere  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

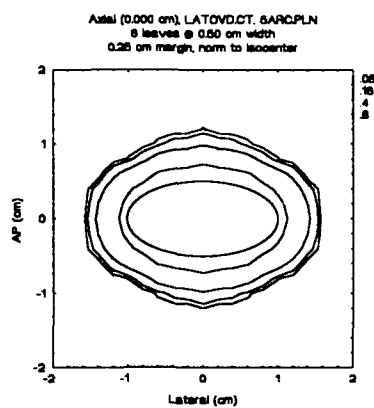


(b)

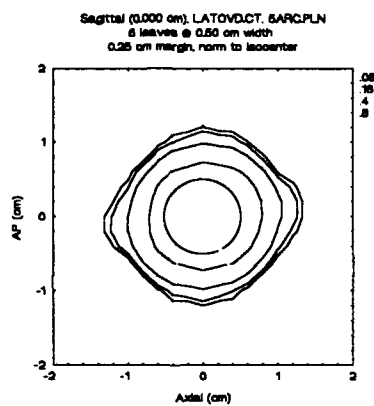


(c)

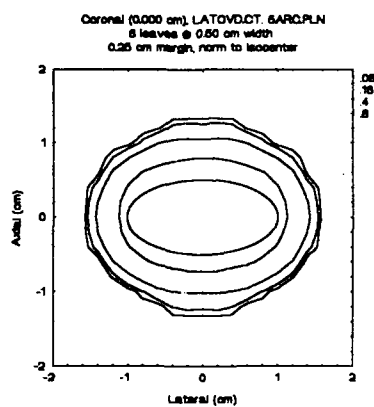
Figure 6-15: Oblique oriented ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

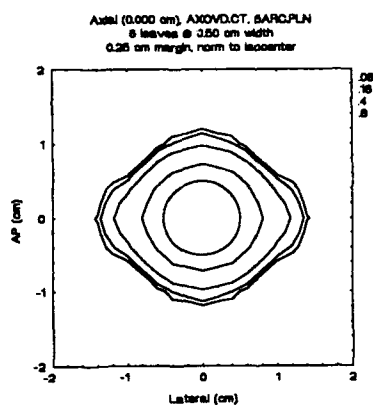


(b)

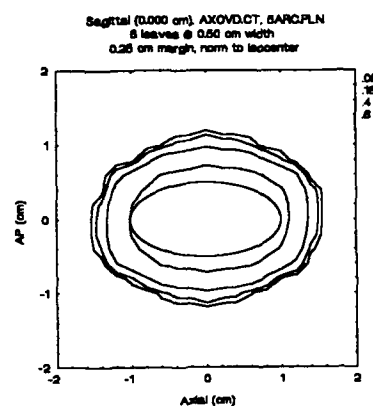


(c)

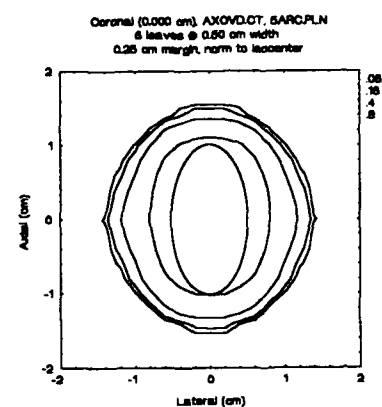
Figure 6-16: Lateral oriented ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

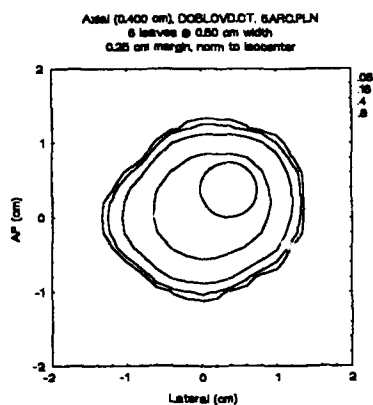


(b)

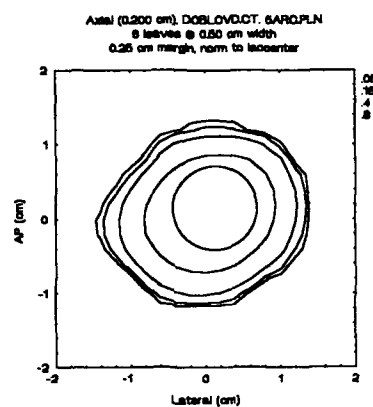


(c)

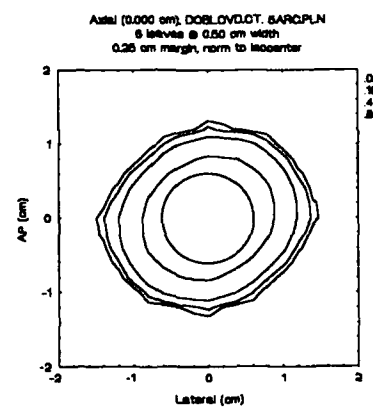
Figure 6-17: Axial oriented ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

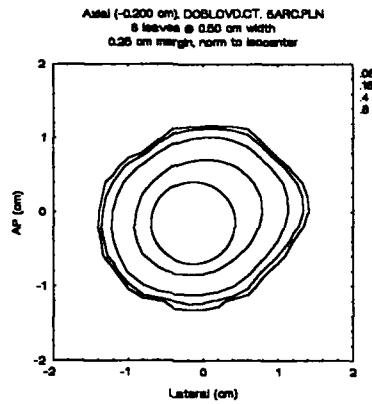


(b)

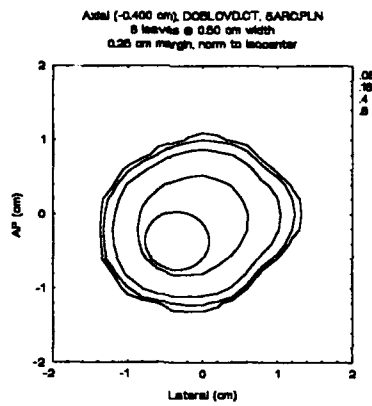


(c)

Figure 6-18: Double oblique oriented ovoid  
(a) Axial +4mm; (b) Axial +2mm; (c) Axial 0mm



(d)



(e)

Figure 6-18 -- continued  
(d) Axial -2mm; (e) Axial -4mm

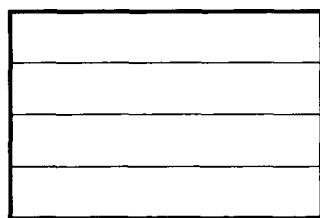
## CHAPTER 7 DOSIMETRY VERIFICATION

The modified negative field method program handles the integration of dynamic irregular fields by the standard numerical method of approximating the integration by a summation of superposited positive and negative fields. As the summation process is trivial, the dosimetry calculation can be verified by demonstrating the correspondence of measured and calculated single static fields. For this demonstration, four irregular field blocks were constructed, placed in the radiation beam, and measured with therapy verification film.

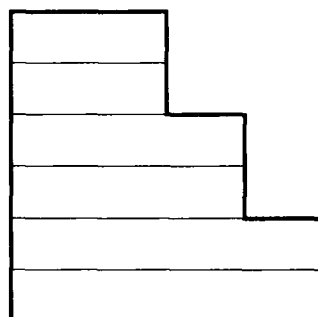
### Irregular Field Block Construction

Four irregular field blocks were constructed of cerrobend. Lucite sheets, 4.6 x 7.6 mm, were cut into strips of 1.0, 1.5, and 2.0 cm and were assembled into prisms with silicon cement. These prisms were the negatives of the planned irregular fields to be constructed and measured.

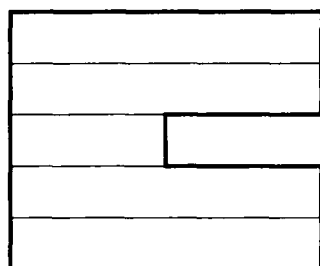
The prisms were placed in the center of square molds and cerrobend was poured around them. This process produced irregular field blocks simulating a multileaf collimator with known leaf width (the lucite thickness, 4.6 mm) and positioning. The block shapes are shown in figure 7-1.



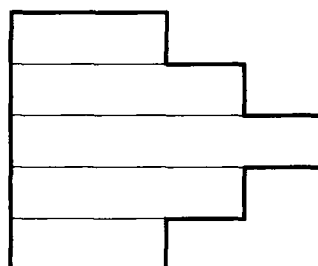
(a)



(b)



(c)



(d)

Note: Prism sections are 4.6 mm x 1.0, 1.5, or 2.0 mm

Figure 7-1: Irregular collimator shapes  
(a) Prism 1; (b) Prism 2; (c) Prism 3; (d) Prism 4



Collimator 1 (figure 7-1a) is built from four 2 cm pieces, resulting in a rectangular collimator with physical dimensions of 2 x 1.8 cm. Collimator 2 (figure 7-1b) is a step collimator built from two pieces each of 1 cm, 1.5 cm, and 2 cm. Collimator 3 (figure 7-1c) is designed to show a single leaf in an open field and is built from 2 cm pieces with a 1 cm piece in the center left. Collimator 4 (figure 7-1d) is a double step with piece dimensions as in collimator 2.

The molds produced non-divergent collimators that were 7.6 cm thick. A solid piece of this thickness was poured and placed in the radiation beam. A transmission factor of 5% was measured for a 10 x 10 cm beam with an ion chamber and this factor was used in the computed dosimetry.

#### Computerized Film Dosimetry

A system was assembled to provide high resolution film dosimetry for verification of the dosimetry programs. The system is centered on an Intel 80386SX microprocessor based personal computer with an 80387SX math coprocessor and Data Translation 2851/2858 video digitizer boards. The boards drive a video camera and a high resolution monitor. The camera is fastened to a vertically moveable mount above a standard light box. Films are placed on the light box, trans-illuminated, and digitized through the computer/frame grabber system.

Software for control of the frame grabber and for operating on buffered image frame data is written in Microsoft C and linked with the Halo 88 device driver library. The software is menu-driven with the following basic options:

(1) Acquire reference films: Reference films are required to calibrate the system to absolute optical density and dose. This is necessary as the video camera is variable in focal length, height above the film, and in aperture. A series of films of known dose and measured optical density (by calibrated film densitometer) are acquired. The optical density measured by the film system is compared to the previously input calibrated optical density and a cubic spline fit is produced to correct the system optical density. The corrected system optical density is further converted to dose by cubic spline interpolation from the reference films originally input.

(2) Flood field correction: The bare light box is digitized and averaged over 50 frames. A multiplier for each pixel in the field is stored based on the maximum illumination found. This may be used as a flood field correction on any images obtained subsequently.

(3) Calibrate distance in x and y axes: Distance must be separately calibrated in both x and y axes to account for asymmetry in pixel coverage (512 pixels on the x axis vs. 480 pixels on the y). A prepared centimeter spaced grid is digitized and known distance points are input by a

mouse-driven cursor on each axis. After calibration, distance between any two points may be directly measured with the cursor. This is also the basis for the distance information included in the output data files to be discussed below.

(4) Acquire film image: The films of interest are then aligned and digitized. Flood field correction (above) may be initiated and multiple frames may be averaged. Three point moving average smoothing may be applied, if requested.

(5) Get image crossplots or areas of interest: Cross sections (in x or y) and/or areas of interest may be defined with the cursor. A center of mass option is available to define initial positioning. The points across or in the defined areas are read from the frame grabber buffer, fit to the established calibrated optical density, and then interpolated to find the corresponding dose. The resulting data is coupled to position, and an ASCII data file of position and dose is written. This data file is structured such that it can be imported into available graphics programs for contour plotting.

#### Irregular Small Field Dose Model Verification

Each collimator was placed in the radiation beam and verification films were exposed to 40 monitor units at a source to film distance of 100 cm under both 1.5 cm ( $d_{\max}$ ) and 10 cm of solid water. Dose calibration films were taken

using a standard 10 x 10 cm field at a source to film distance of 100 cm and a depth of  $d_{\text{max}}$ , with exposures of 0, 20, 40, and 60 monitor units to give film doses of 0, 20, 40, and 60 centigray. The collimator films produced at a depth of 10 cm were digitized by the computerized film dosimetry system and the resulting images were stored as two dimensional matrices for display as isodose plots. Cross plots of each image were obtained manually with the MacBeth film densitometer as a second check.

Each collimator dimension was then written into a localization file and was processed by the plane dosimetry program (appendix C). The output was saved for comparison, each isodose plot being compared on the 30%, 50%, and 70% isodose lines. Calculated cross plots corresponding to the measured cross plots were also obtained, the axial or lateral off axis distances being referenced to the coordinates of the isodose plots.

Table 7-1: Isodose plot average maximum position errors

Isodose line	30%	50%	70%
Collimator 1	0.67 mm	0.34 mm	0.43 mm
Collimator 2	0.31 mm	0.20 mm	0.26 mm
Collimator 3	0.24 mm	0.25 mm	0.51 mm
Collimator 4	0.30 mm	0.18 mm	0.17 mm

The isodose plots were prepared mainly for qualitative evaluation, i.e. to determine if the dosimetry program produces a proper isodose curve shape. Quantitative evalua-

tion was minimal. Isodose line variation was measured by overlaying the isodose plots and finding the maximum error on each side. These errors were averaged to find the average maximum position error (table 7-1).

Area integration was done on the cross plots (using Easy Plot version 2.2, Spiral Software, Brookline MA), with the square root of the absolute percent error of the calculated with respect to the measured values being taken as an average error per point over the line. This was performed for all points above the 40% normalized dose line, the high dose area (table 7-2).

Table 7-2: Cross plot average error per point

Depth	1.5 cm	10 cm
Collimator 1 (lateral)	1.47%	0.82%
Collimator 1 (axial)	1.64%	1.19%
Collimator 2 (lateral)	0.80%	1.45%
Collimator 2 (axial)	0.43%	1.36%
Collimator 3 (lateral)	2.96%	1.24%
Collimator 3 (axial)	2.90%	1.57%
Collimator 4 (lateral)	1.78%	1.89%
Collimator 4 (axial)	1.69%	2.83%

The maximum error for any single point in each cross plot was measured for both dose (on the plateau) and for position (on the slope) for the high dose ( $\geq 40\%$  dose) and for the low dose ( $< 40\%$  dose) areas. These errors are listed in table 7-3.

The contour plots of collimator 1 (figure 7-2) show good agreement between the measured and the calculated

Table 7-3: Cross plot maximum point errors

Depth	1.5 cm		10 cm	
Dose line	$\geq 40\%$	$< 40\%$	$\geq 40\%$	$< 40\%$
Collimator 1 (axial)	-2.1% -0.3 mm	-1.5% -0.5 mm	-1.5% -0.2 mm	-5.5% -1.2 mm
Collimator 1 (lateral)	-2.0% -0.4 mm	-2.6% -0.7 mm	-2.1% -0.5 mm	-4.6% -1.1 mm
Collimator 2 (axial)	-1.2% -0.4 mm	-1.4% -0.3 mm	-2.3% -0.6 mm	-6.0% -0.9 mm
Collimator 2 (lateral)	-1.2% -0.4 mm	-2.2% -0.4 mm	-2.9% -0.5 mm	-4.1% -1.1 mm
Collimator 3 (axial)	-1.2% -0.3 mm	-2.3% -0.5 mm	-2.1% -0.4 mm	-4.1% -1.0 mm
Collimator 3 (lateral)	-1.2% -0.4 mm	-2.0% -0.6 mm	-2.9% -0.4 mm	-4.6% -1.3 mm
Collimator 4 (axial)	-2.8% -0.4 mm	-3.2% -0.7 mm	-3.5% -0.6 mm	-3.8% -0.8 mm
Collimator 4 (lateral)	-3.2% -0.5 mm	-2.1% -0.5 mm	-3.8% -0.2 mm	-4.6% -0.6 mm

isodose lines. The cross plots (figures 7-3 and 7-4) agree. The average maximum position error between the isodose plots is small as are the average error and the maximum error per point on the cross plots.

The contour plots of collimator 2 (figure 7-5) show the same good agreement. The cross plots (figures 7-6 and 7-7) also agree. The average maximum position error between the isodose plots is small as are the average error and the maximum error per point on the cross plots.

The contour plots of collimator 3 (figure 7-8) also show good agreement. The cross plots (figures 7-9 and 7-10) again agree. The average maximum position error between the isodose plots is small as are the average error and maximum error per point on the cross plots.

The contour plots of collimator 4 (figure 7-11) continue the same good agreement, as do the cross plots (figures 7-12 and 7-13). The average maximum position error between the isodose plots is small as are the average error and the maximum error per point on the cross plots.

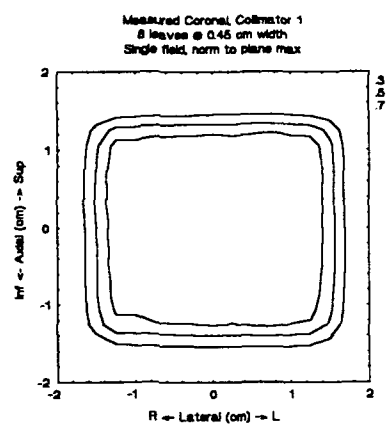
#### Analysis Results

The position errors (table 7-1) are on the order of the CT pixel resolution used in treatment planning [Fri89b]. The average cross plot errors (table 7-2) are all less than 3%, and in most cases are considerably less. The overall average of these errors is 1.63%. The maximum point errors for the high dose region peak at -3.2% and -0.5 mm (average -1.9% and -0.4 mm) for 1.5 cm depth and -3.8% and -0.6 mm (average -2.6% and -0.4 mm) for 10 cm depth. The maximum point errors for the low dose region, of lesser importance, peak at -3.2% and -0.7 mm (average -2.2% and -0.5 mm) for

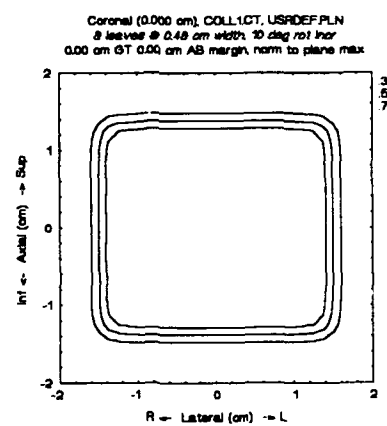
1.5 cm depth and -6.0% and -1.2 mm (average -4.7% and 1.0 mm) for 10 cm depth.

The good fit of the calculated dosimetry to the measured films is confirmed. The modified negative field method is shown to provide acceptable dosimetry for stereotactic radiosurgery using irregular, conformal fields.



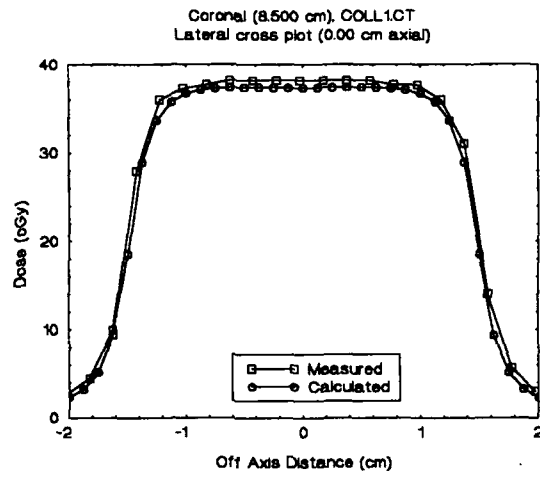


(a)

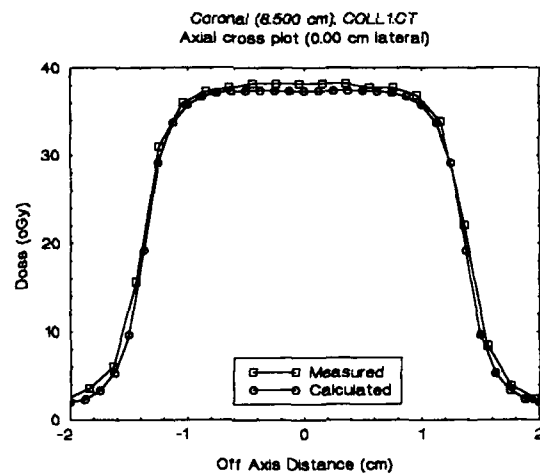


(b)

Figure 7-2: Collimator 1 isodose plots  
(a) Measured; (b) Calculated

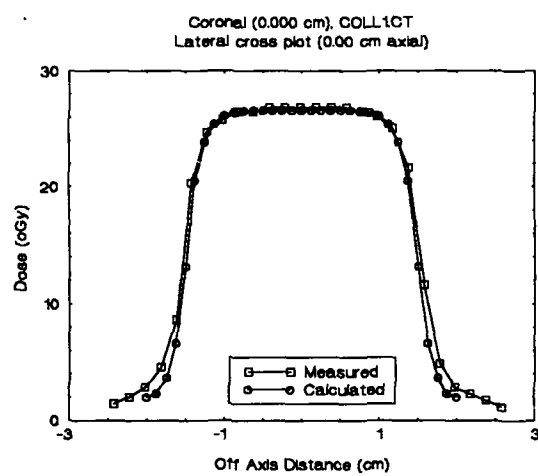


(a)

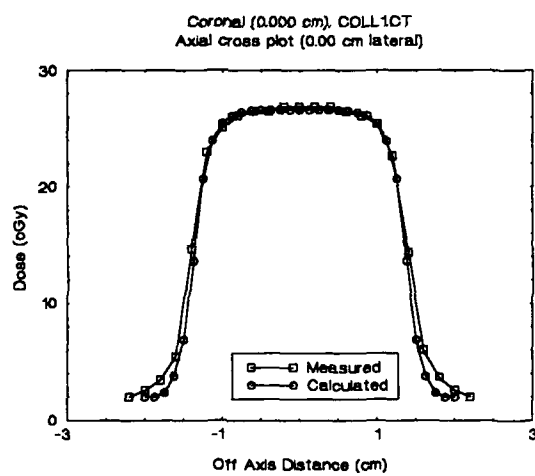


(b)

Figure 7-3: Collimator 1 cross plots, 1.5 cm depth  
(a) Lateral; (b) Axial

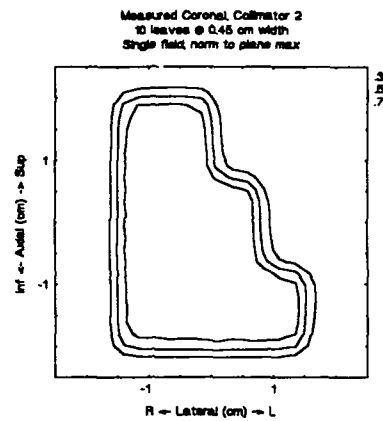


(a)

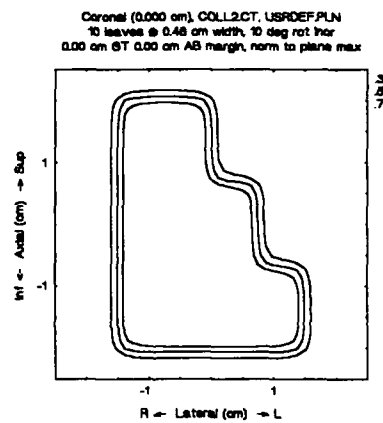


(b)

Figure 7-4: Collimator 1 cross plots, 10 cm depth  
(a) Lateral; (b) Axial

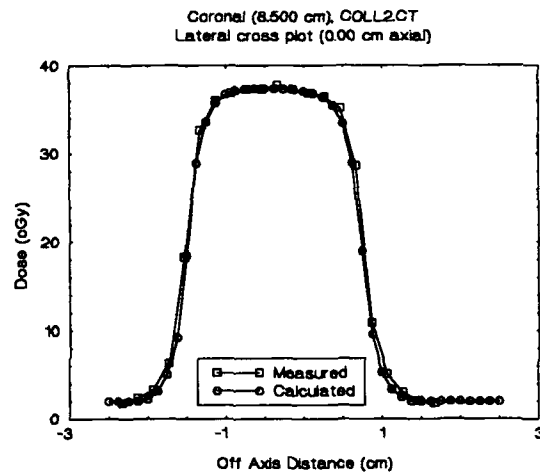


(a)

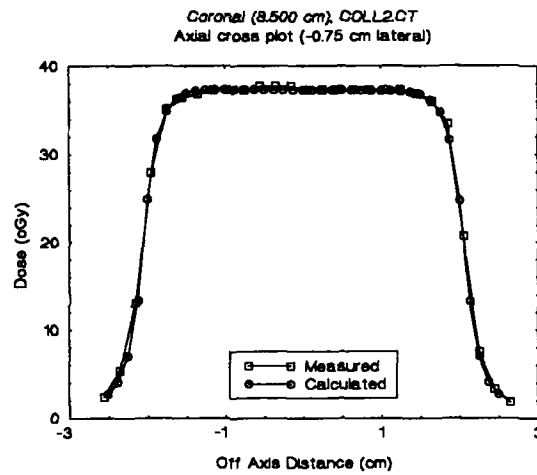


(b)

Figure 7-5: Collimator 2 isodose plots  
(a) Measured; (b) Calculated

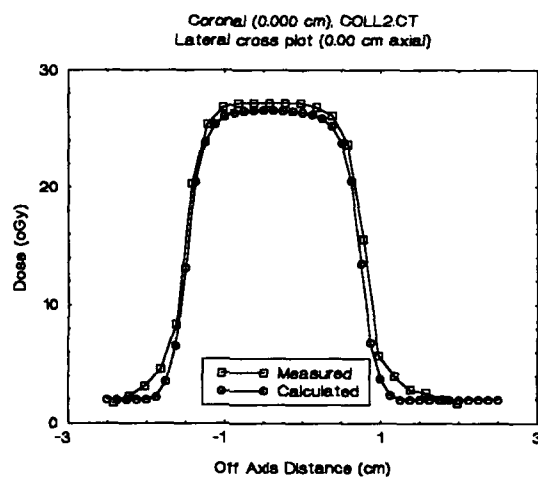


(a)

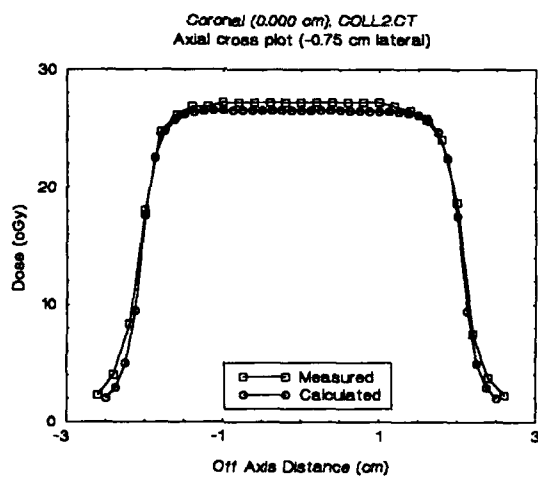


(b)

Figure 7-6: Collimator 2 cross plots, 1.5 cm depth  
(a) Lateral; (b) Axial

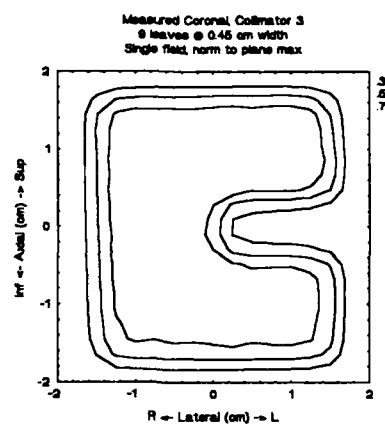


(a)

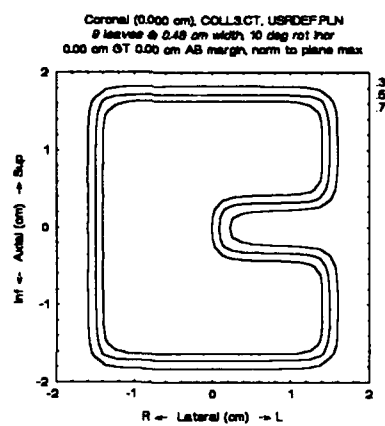


(b)

Figure 7-7: Collimator 2 cross plots, 10 cm depth  
(a) Lateral; (b) Axial

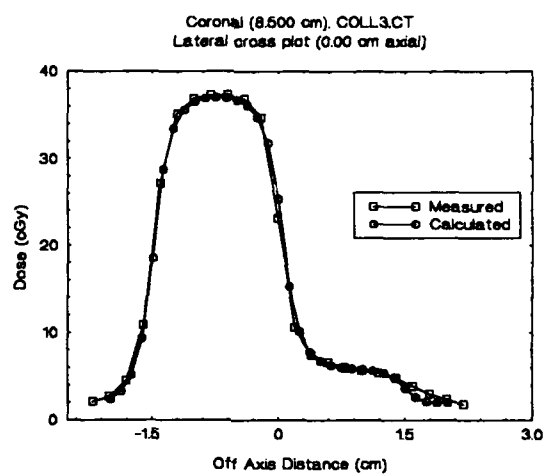


(a)

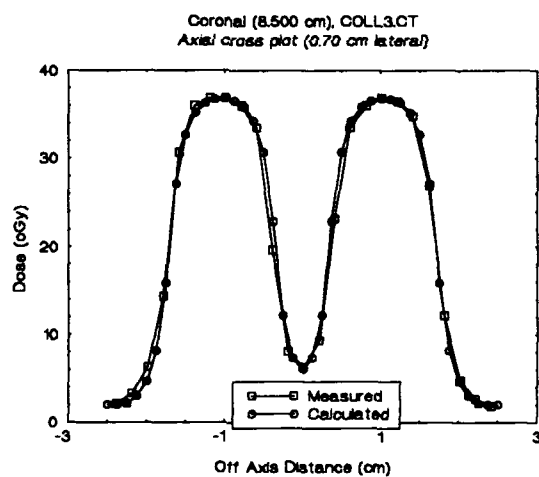


(b)

Figure 7-8: Collimator 3 isodose plots  
(a) Measured; (b) Calculated



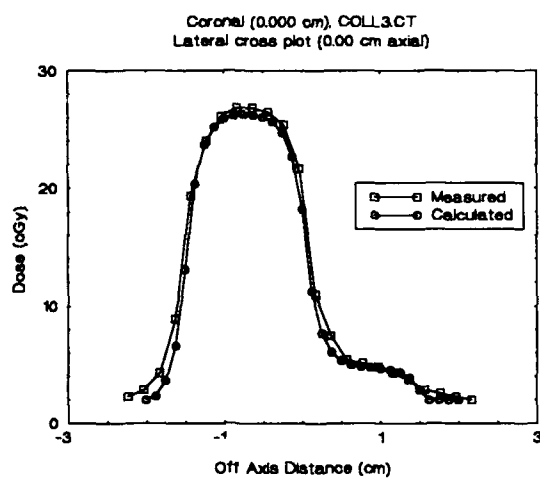
(a)



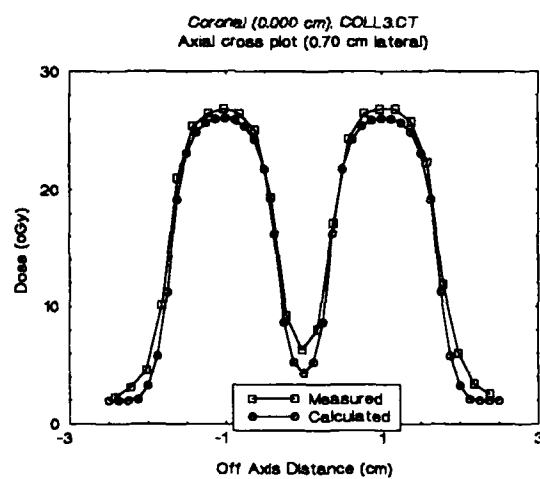
(b)

Figure 7-9: Collimator 3 cross plots, 1.5 cm depth  
(a) Lateral; (b) Axial



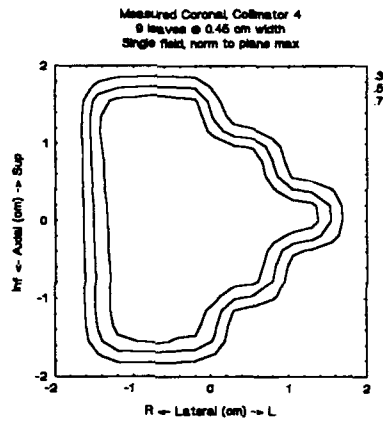


(a)

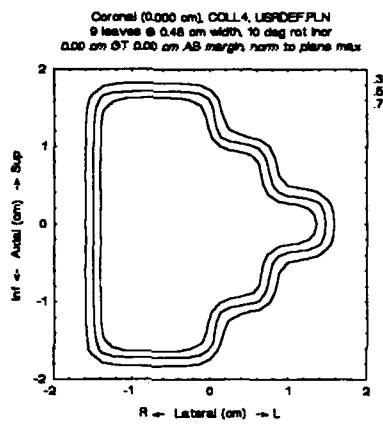


(b)

Figure 7-10: Collimator 3 cross plots, 10 cm depth  
(a) Lateral; (b) Axial

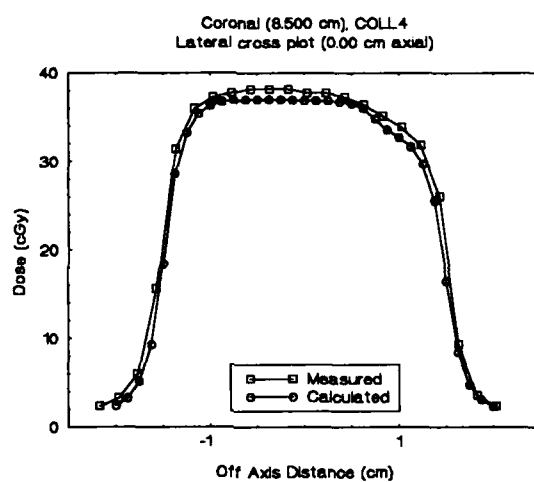


(a)

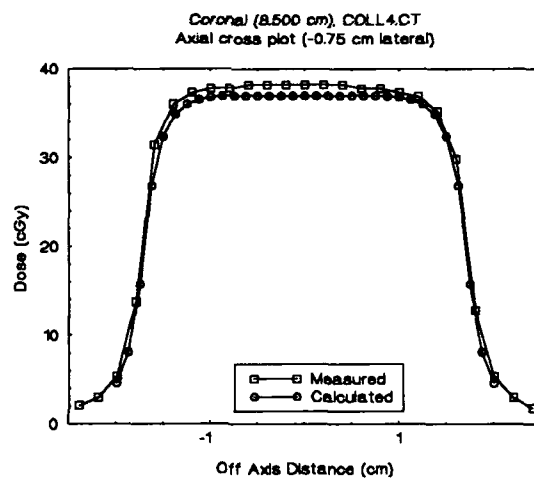


(b)

Figure 7-11: Collimator 4 isodose plots  
(a) Measured; (b) Calculated

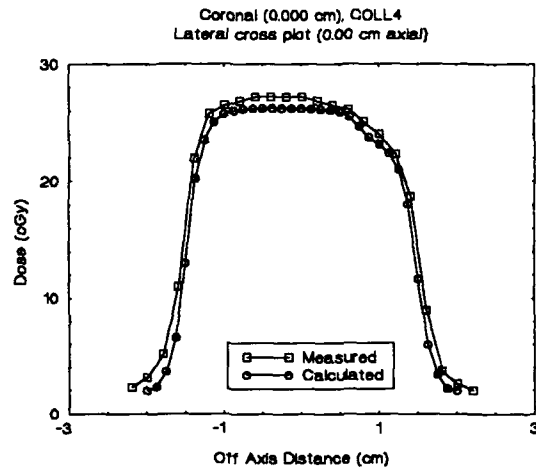


(a)

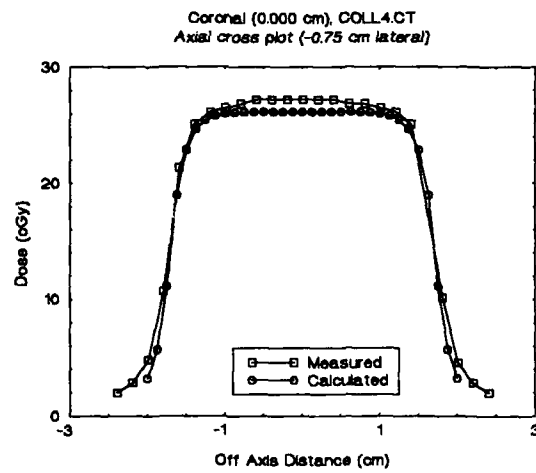


(b)

Figure 7-12: Collimator 4 cross plots, 1.5 cm depth  
(a) Lateral; (b) Axial



(a)



(b)

Figure 7-13: Collimator 4 cross plots, 10 cm depth  
(a) Lateral; (b) Axial

## CHAPTER 8 COLLIMATOR SPECIFICATION

The design features of a small field multileaf conformal collimator for the University of Florida Stereotactic Radiosurgery System will be discussed and specified. The specification will include leaf shape and optimal leaf width for both two jaw and four jaw localization (previously seen to be ideal). Also shown will be the effects of coronal arc compression, gantry incrementation, and localization margin on dose distributions.

### Leaf Shape

Most, if not all, field blocks used for large field teletherapy have the blocked area tapered to match the divergence of the radiation beam. This improves beam penumbra by eliminating partial block transmission effects. In small field blocking (fields of less than about 3 cm square at the position of the block) the beam divergence is also quite small (on the order of  $1^\circ$  from the beam central axis). This small divergence should have a minimal effect on the beam at the block edge.

To test this assertion, films were made of the field edge under a straight edged lead block, 5 cm thick, and under the same block angled with spacers to match the beam

divergence. The films were read with a MacBeth densitometer and the optical density was plotted against the position. As shown in figure 8-1, there is no significant difference between the edge effects of the straight edged block when compared to the tapered.

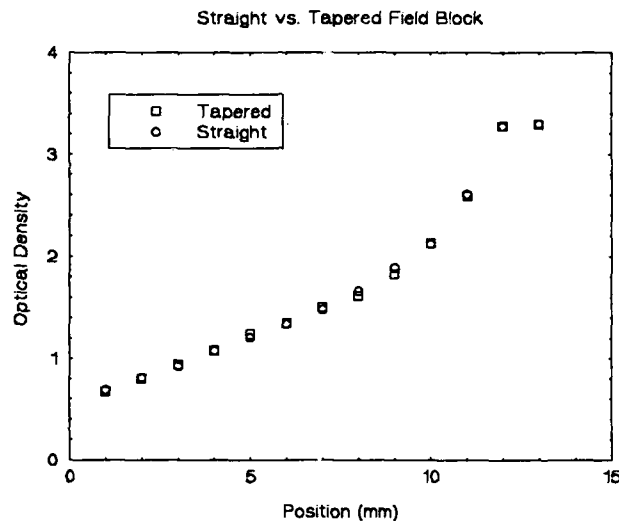


Figure 8-1: Straight vs. tapered beam blocking

This greatly simplifies the design of the collimator, as collimators with blocks made to follow beam divergence with the block off axis position are complex.

#### Leaf Width

The effects of leaf width are shown in figures 8-2 through 8-6 for the AP ovoid target, figures 8-7 through 8-11 for the spherical target, 8-12 through 8-16 for the oblique ovoid target, figures 8-17 through 8-21 for the lateral ovoid, figures 8-22 through 8-26 for the axial ovoid, and figures 8-27 through 8-31 for the double oblique

ovoid, for leaf widths of 2.5 mm, 5 mm, 10 mm, and 30 mm. The standard 5 arc treatment plan previously defined was used for this dosimetry.

The isodose contours on the principal planes for each target show a consistent spreading as the leaf width increases. This is presumably due to the decreasing goodness of fit of the collimator leaves to the target as the leaf width increases, which leads to an increase in the average collimator aperture. The increase is most notable for the widths larger than 5 mm. This observation is confirmed by observing the dose volume histograms for each target. These show an increase in both the cumulative dose given to the target and to the normal tissue, indicating an over coverage of the target as posited above.

Plotting an integrated logistic function complication probability on each of the set of dose volume histograms for each separate target shows this process more clearly (figures 8-32 through 8-37). In each case the probability increases with increasing leaf width, indicating that the cumulative dose to normal tissue increases similarly. These plots also show a steady decrease in probability improvement as leaf width increases (recall that a lower probability number is defined as "better" for normal tissue). Given that the time for dose computation increases with decreasing leaf width, each leaf forming a separate negative field in the dosimetry model, using a width of 5 mm gives both a good

dosimetric fit and is in the approximate center of the diminishing return floor of the integrated logistic function probabilities.

#### Arc Compression

Conformal dosimetry depends on the superposition of many discrete conformal fields that are distributed around the target. The best distribution, conceptually, is a  $4\pi$  distribution that illuminates the target from all directions. This is obviously physically impossible for a human undergoing stereotactic treatment. The next best distribution, or the best distribution physically possible, is a  $2\pi$  coronal cephalad distribution. This distribution is approximated by the standard 5 through 9 arc stereotactic treatment plans used by the University of Florida stereotactic radiosurgery system. In circular field (i.e. standard) radiosurgery, frequently the lateral arcs are removed to increase the axial spread of the dose distribution to cover axially extended targets. This is a physical process which figure 8-38 shows is not compromised in conformal dosimetry, and is in fact inimical to the achievement of a good dosimetric fit. Comparing figure 5-38 with any of the other plane dosimetry figures shows that a  $2\pi$  arc distribution is optimal in conformal dosimetry and treatment.

#### Gantry Incrementation

As stated previously, conformal dosimetry employs the superposition of many discrete conformed fields distributed



around the target. Computationally, the only variable available to control the number of superposited fields is the increment of gantry rotation. Figures 8-39 through 8-41, for the AP ovoid, and figures 8-42 through 8-44, for the sphere, show the effects of increasing the gantry rotation increment from  $5^\circ$  to  $10^\circ$  to  $20^\circ$ , successively. The  $5^\circ$  and  $10^\circ$  increments show no discernable differences. The  $10^\circ$  and  $20^\circ$  increments show a slight, though noticeable, degradation in quality and smoothness. As computation time increases with a decrease in gantry increment, the  $10^\circ$  increment, which decreases computation time with no decrease in the quality of the dosimetry, is used for all subsequent calculations.

#### Localization Margin

Best fit of isodose lines to the target periphery is controlled by the margin of localization (i.e the closest approach of each leaf to the target) about each projected target area at each increment of gantry rotation. Several hundred plans were run and evaluated to find the empirical optimal best fit of the 80% isodose line to the target in each of the three principal planes (axial, sagittal, and coronal) for four jaw localization. The evaluation, empirical as noted, was of the visual optimization method currently employed for multiple isocenter plan optimization. These plans are illustrated in figure 8-45 for the sphere, figure 8-46 for the AP ovoid, figure 8-47 for the lateral ovoid,

figure 8-48 for the axial ovoid, figure 8-49 for the oblique ovoid, and figure 8-50 for the double oblique ovoid. All distributions show a good fit to the target with a margin of 1 mm on each collimator side.

This margin was used to illustrate the simpler design problem of a collimator localizing the target with two jaws on the A and B sides of the gantry. The leaf width was halved to 2.5 mm for these plans. These distributions are shown in figure 8-51 for the sphere, figure 8-52 for the AP ovoid, figure 8-53 for the lateral ovoid, figure 8-54 for the axial ovoid, figure 8-55 for the oblique ovoid, and figure 8-56 for the double oblique ovoid. All targets are fit to the 80% line. As previously noted in chapter 5, the higher isodose lines broaden slightly and the lower lines broaden significantly with two jaw localization, however this may be considered against the favorable tradeoff of the noted simpler collimator design.

#### Conclusion

The design factors for the realization of a small field multileaf collimator are the leaf shape (tapered or straight), the leaf width, and the localization margin. Other properties directly related to the collimator and the resulting dosimetry are the arc distribution and the gantry incrementation.

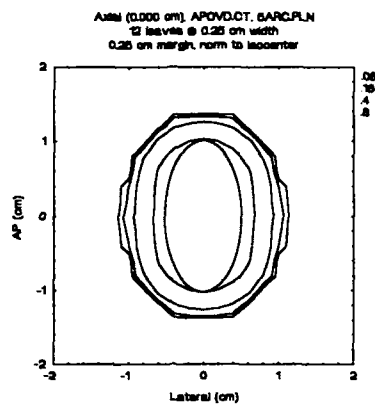
The leaf shape has been shown to be insensitive to the angles of beam divergence encountered in small fields. A

straight edged leaf is preferable and is acceptable from the design standpoint.

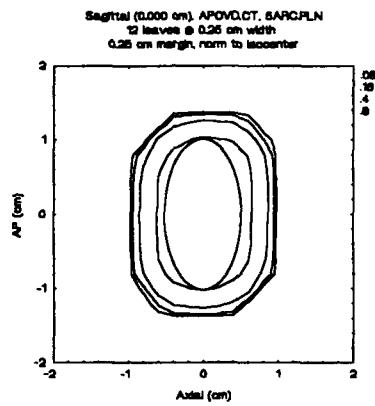
The effect of leaf width on the dosimetry has been investigated by principal plane dosimetry, by dose volume histogram, and by applying and comparing the results of the integrated logistic function. The width found to be best in terms of fit and computation time is 5 mm for four jaw localization.

Localization margin was exhaustively investigated by comparing the results of many principal dosimetry plane plots searching for the best fit of the target to the 80% isodose line. The empirical best fit was found with a constant margin of 1 mm on each target, and again the superiority of four jaw vice two jaw localization was demonstrated, with the not unimportant qualification that the design and implementation of a two jaw collimator is much simpler than that of the four jaw.

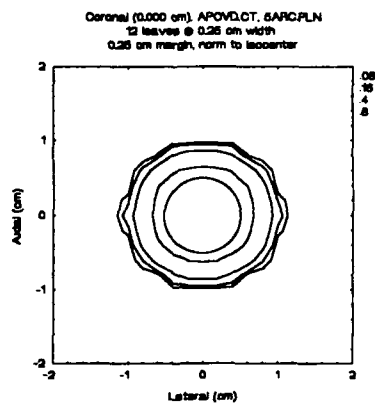
Of the other properties, the arc distribution has been shown to be optimal when approximating a  $2\pi$  coverage. A gantry incrementation of  $10^\circ$  has been shown to be preferable based on computation time.



(a)

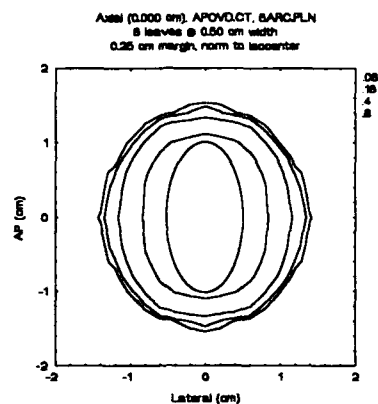


(b)

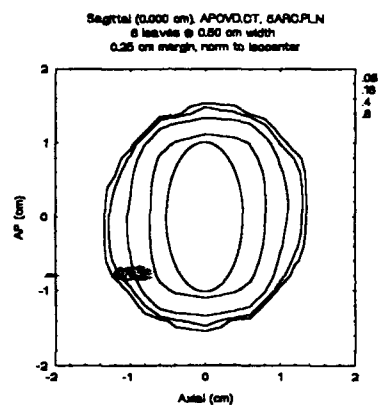


(c)

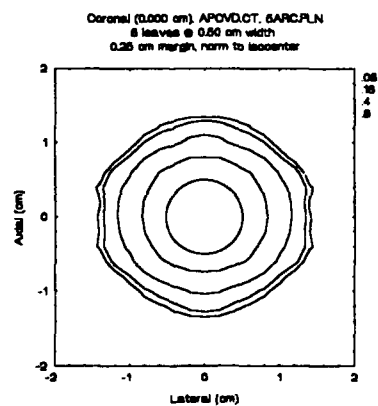
Figure 8-2: AP ovoid, 2.5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

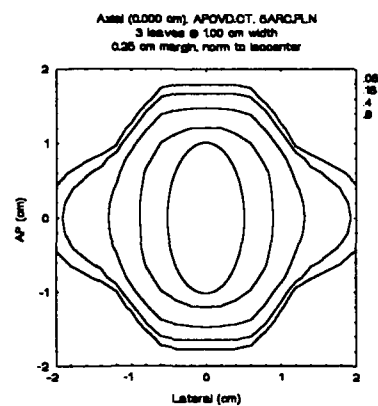


(b)

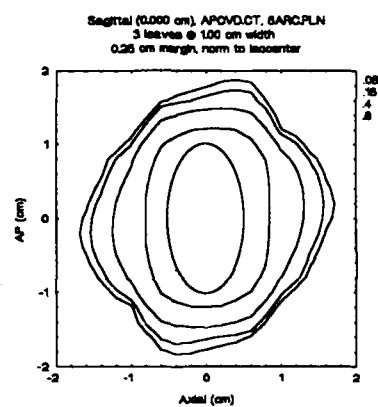


(c)

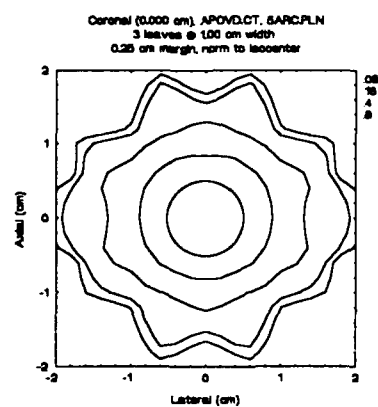
Figure 8-3: AP ovoid, 5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

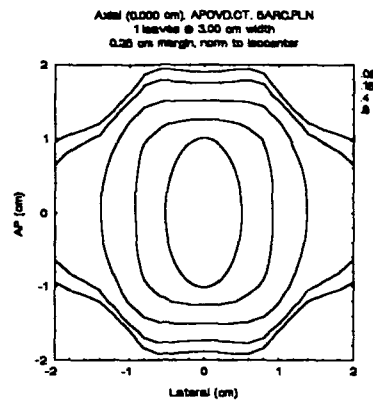


(b)

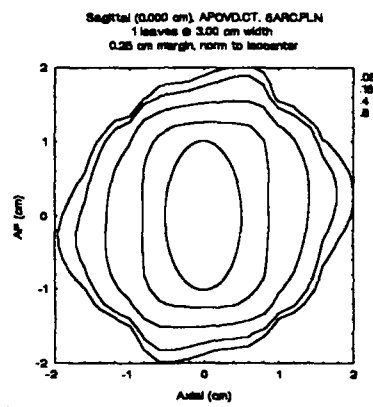


(c)

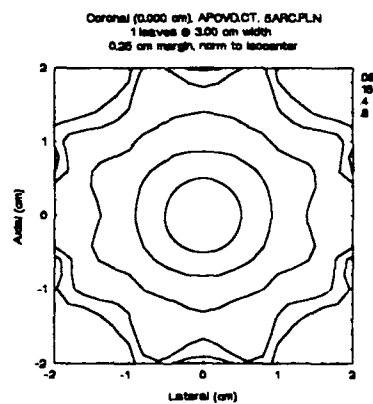
Figure 8-4: AP ovoid, 10 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

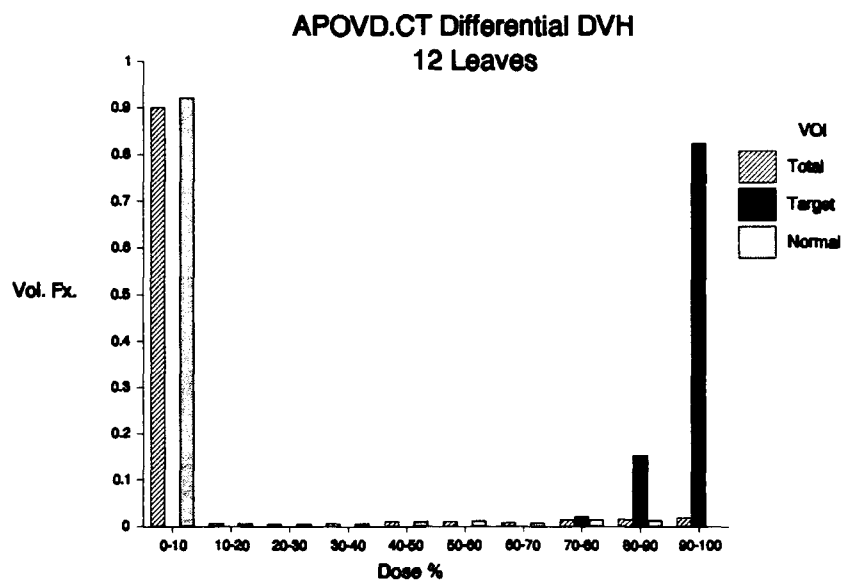


(b)

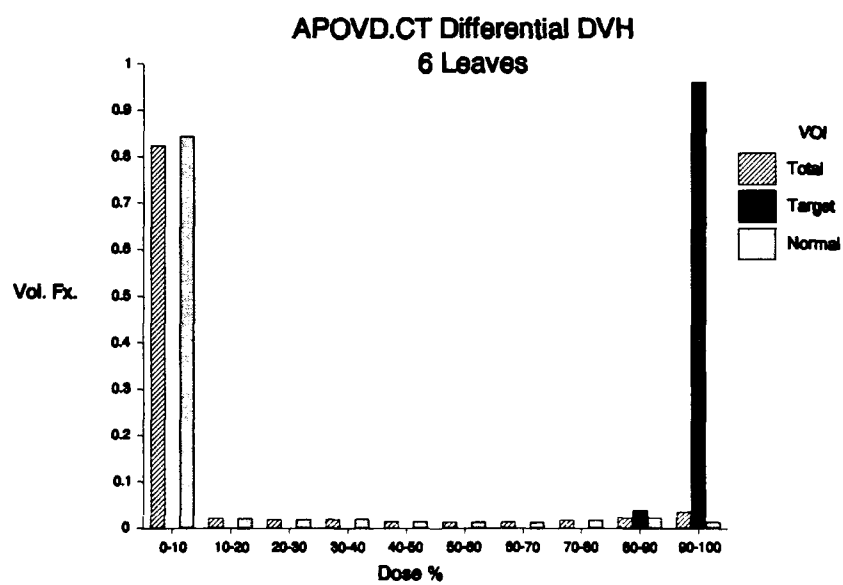


(c)

Figure 8-5: AP ovoid, 30 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



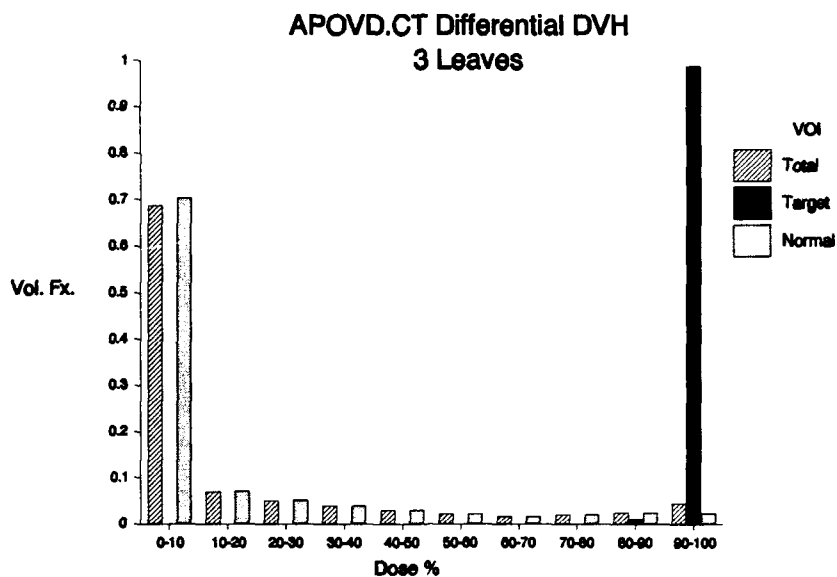
(a)



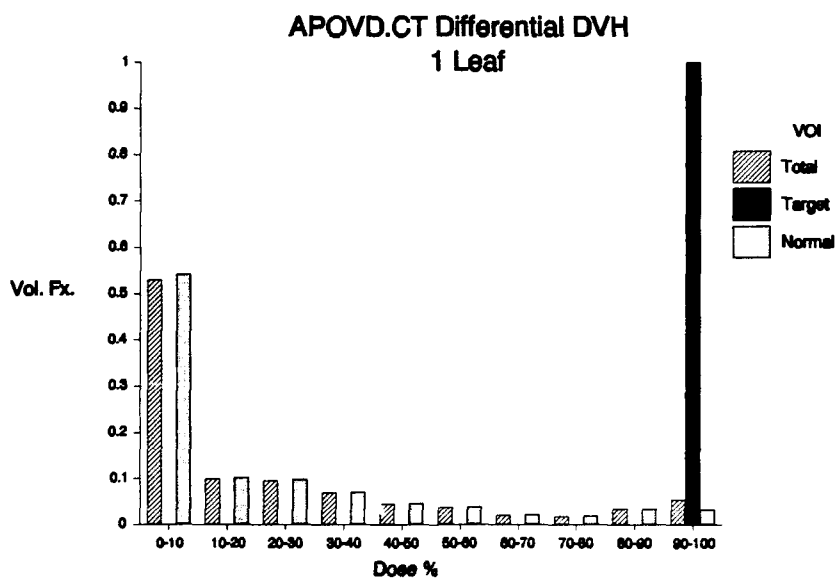
(b)

Figure 8-6: AP ovoid dose volume histogram  
(a) 2.5 mm leaf localization; (b) 5 mm leaf localization



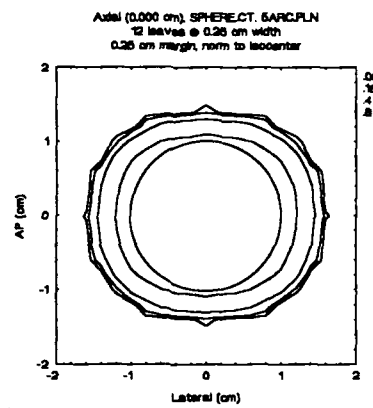


(c)

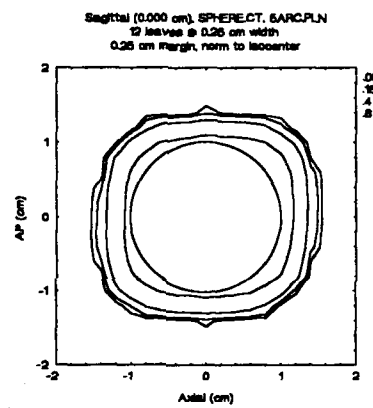


(d)

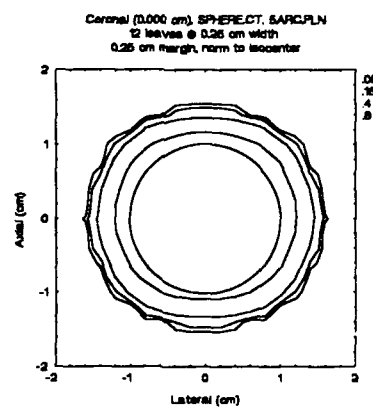
Figure 8-6 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization



(a)

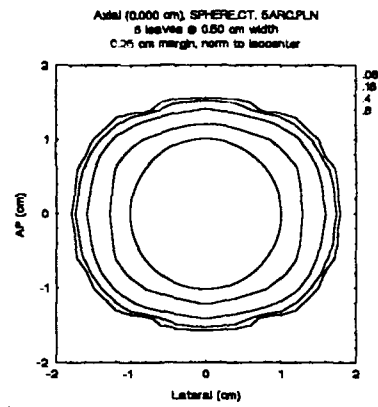


(b)

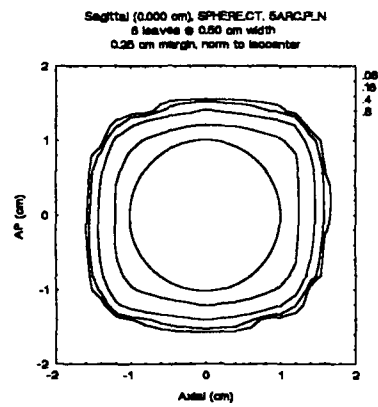


(c)

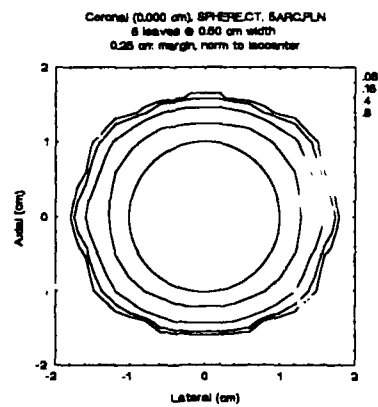
Figure 8-7: Sphere, 2.5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

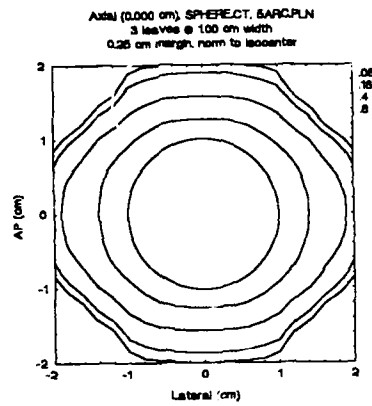


(b)

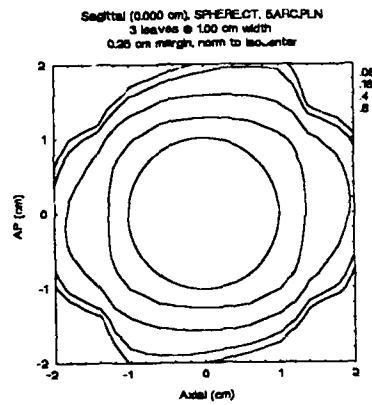


(c)

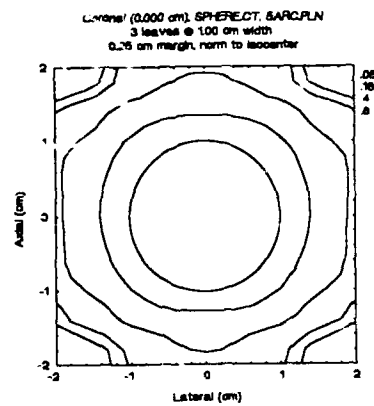
Figure 8-8: Sphere, 5 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

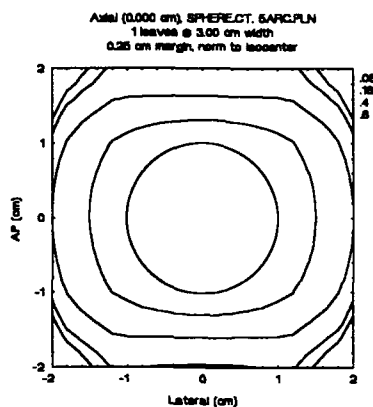


(b)

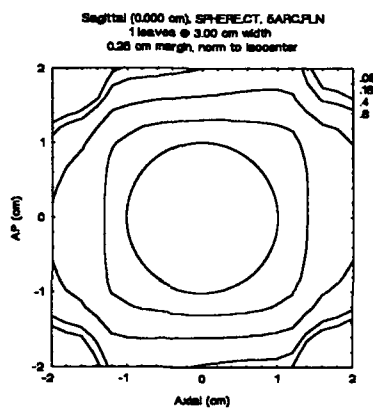


(c)

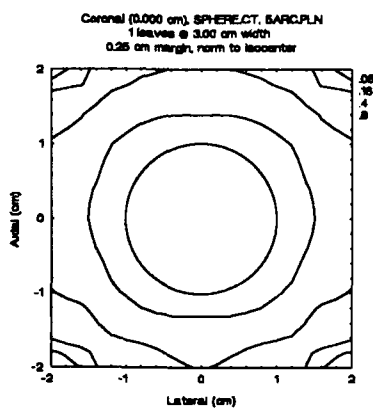
Figure 8-9: Sphere, 10 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

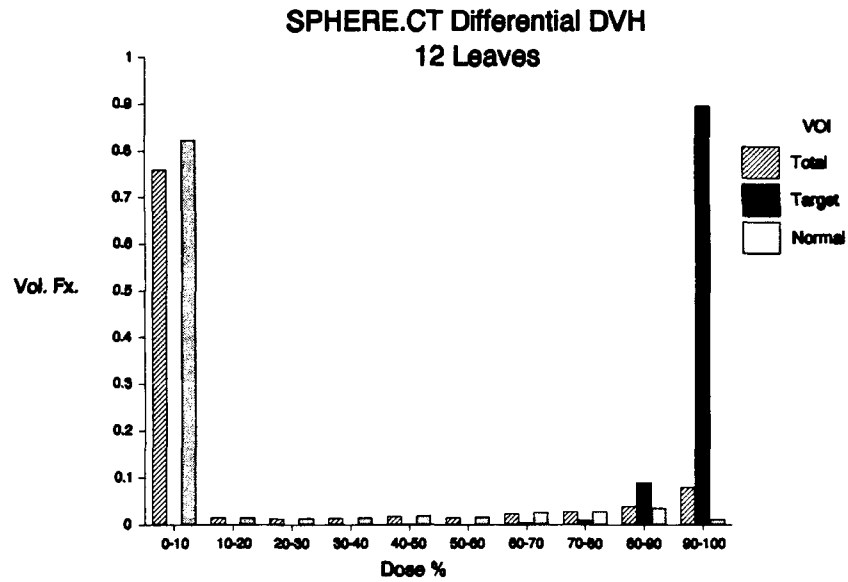


(b)

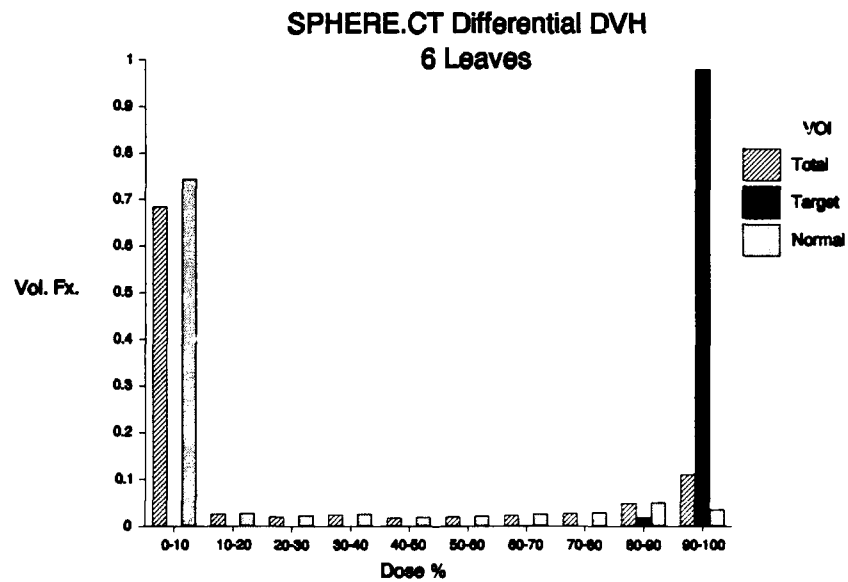


(c)

Figure 8-10: Sphere, 30 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut

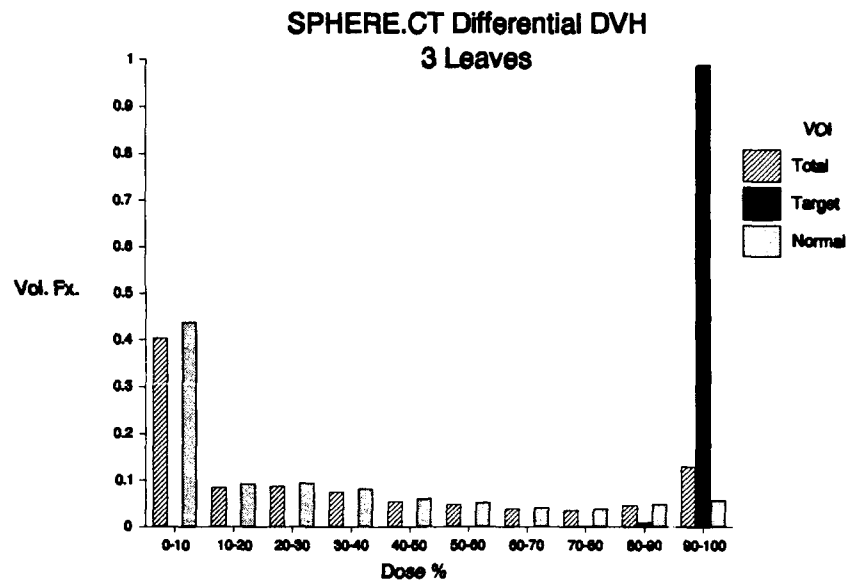


(a)

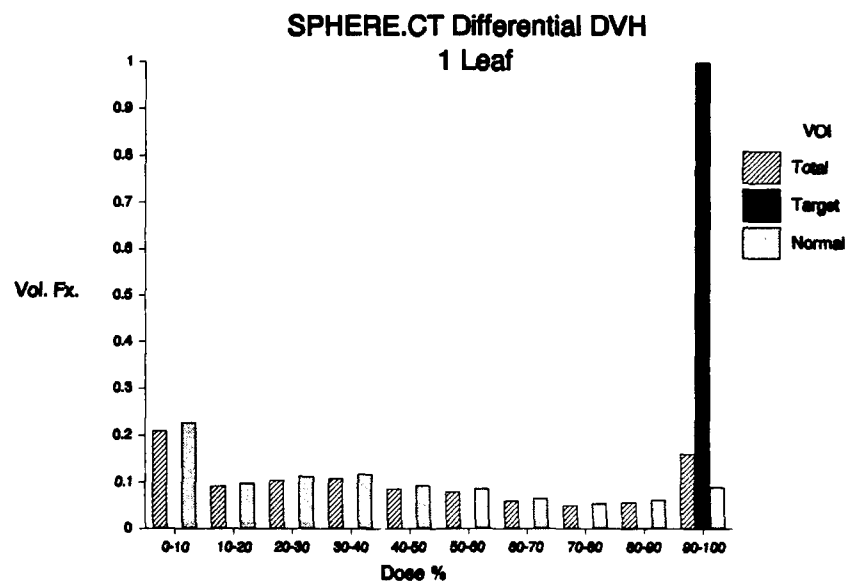


(b)

Figure 8-11: Sphere dose volume histogram  
(a) 2.5 mm leaf localization; (b) 5 mm leaf localization

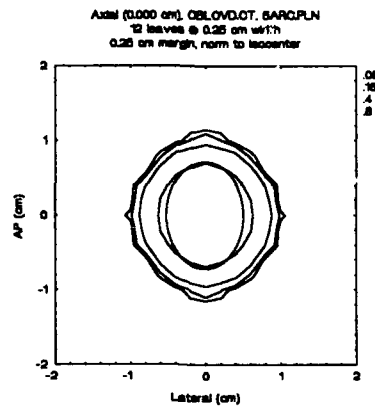


(c)

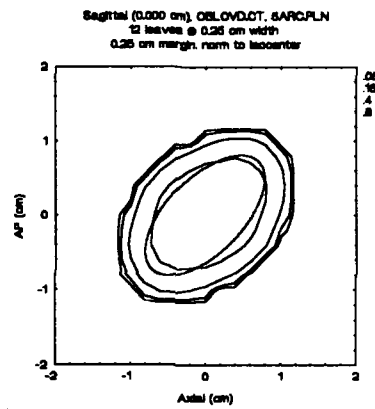


(d)

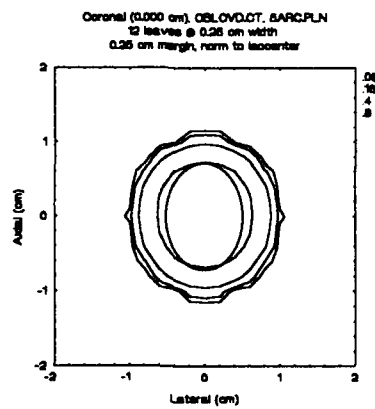
Figure 8-11 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization



(a)



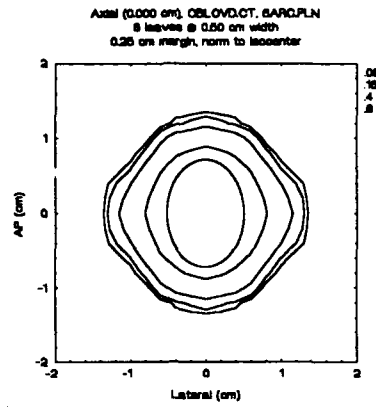
(b)



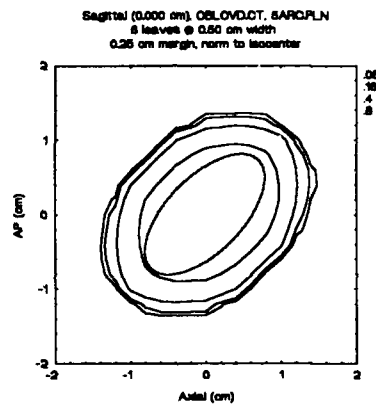
(c)

Figure 8-12: Oblique ovoid, 2.5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

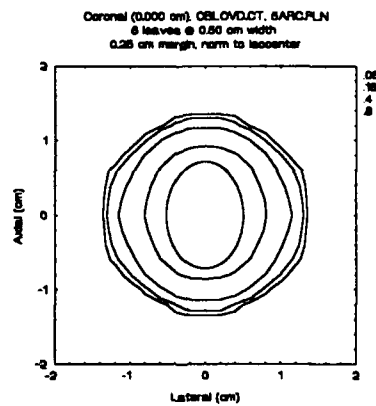




(a)

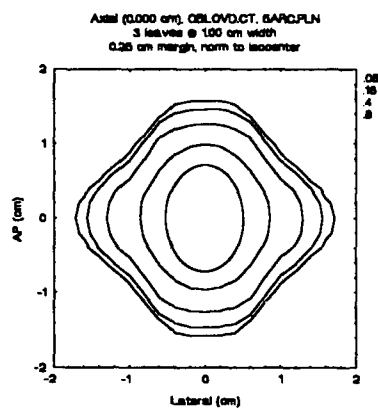


(b)

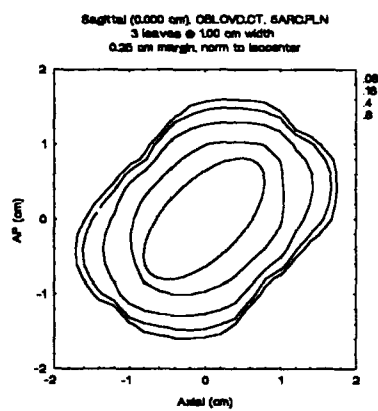


(c)

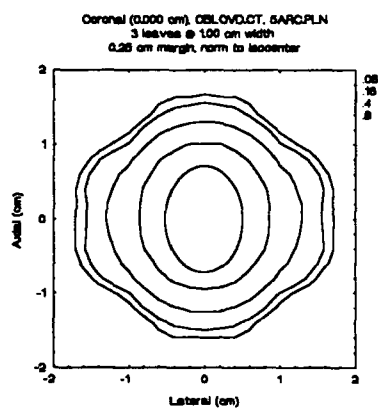
Figure 8-13: Oblique ovoid, 5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

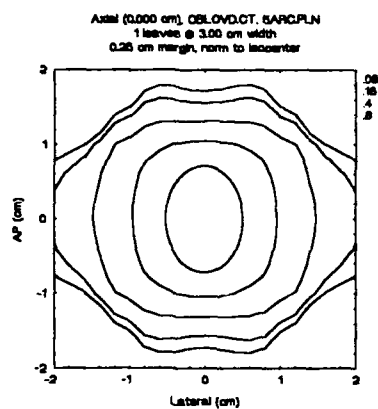


(b)

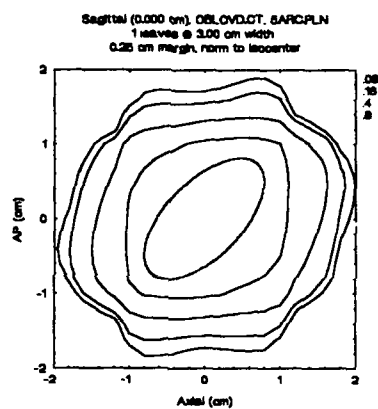


(c)

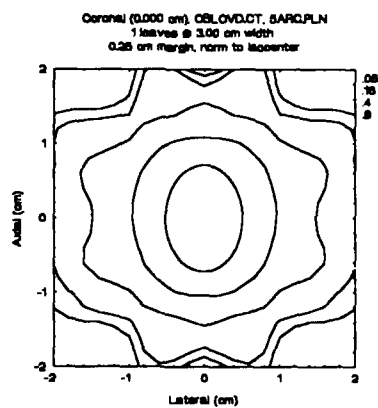
Figure 8-14: Oblique ovoid, 10 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

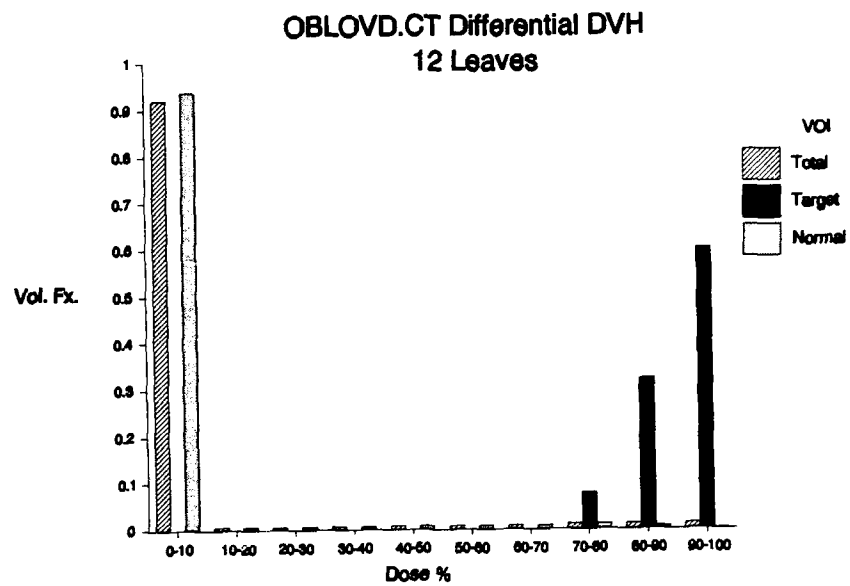


(b)

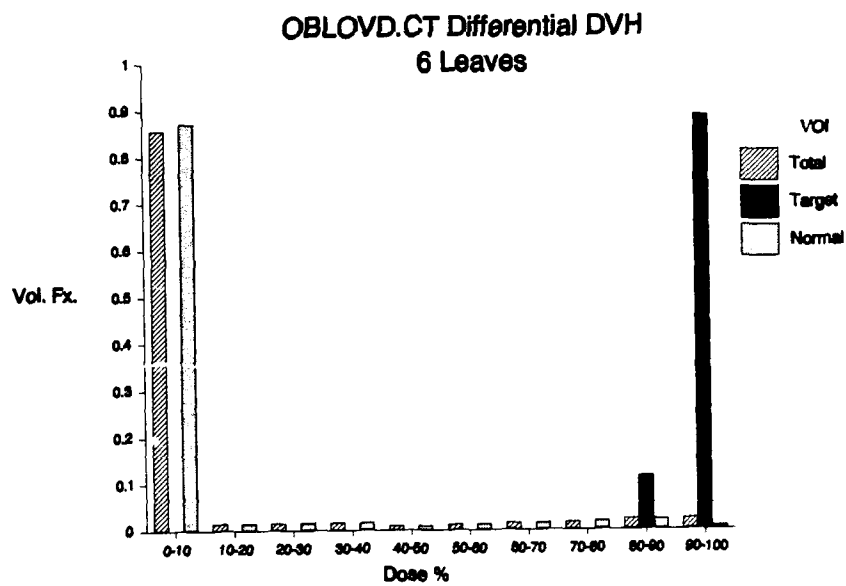


(c)

Figure 8-15: Oblique ovoid, 30 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut

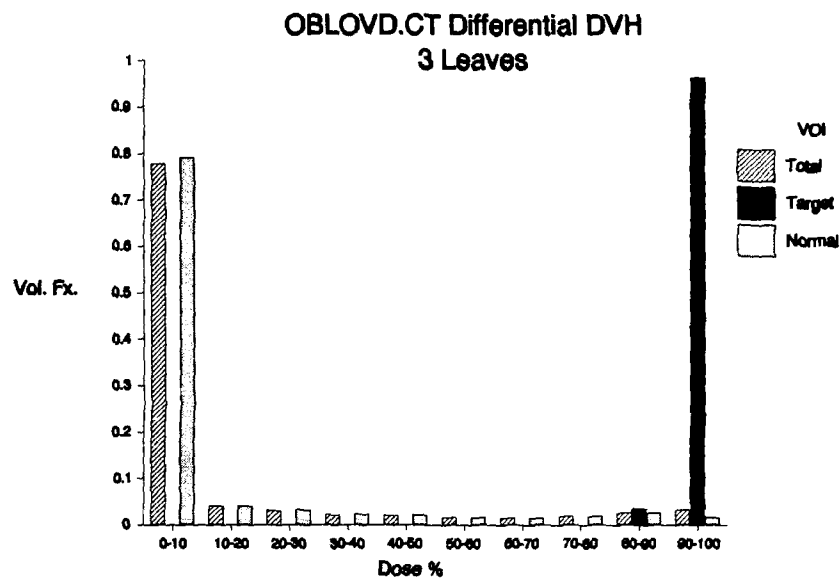


(a)

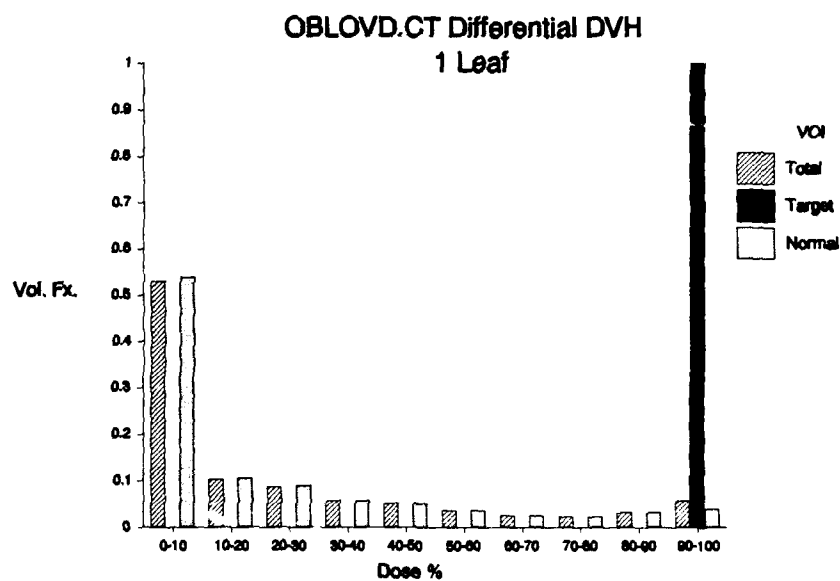


(b)

Figure 8-16: Oblique ovoid dose volume histogram  
(a) 2.5 mm leaf localization; (b) 5 mm leaf localization

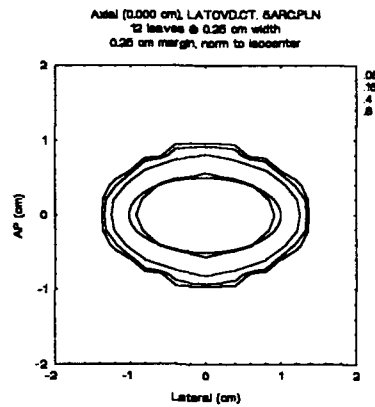


(c)

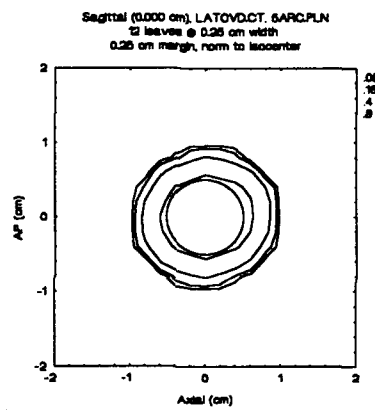


(d)

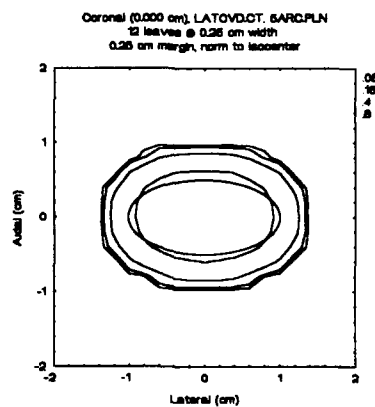
Figure 8-16 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization



(a)

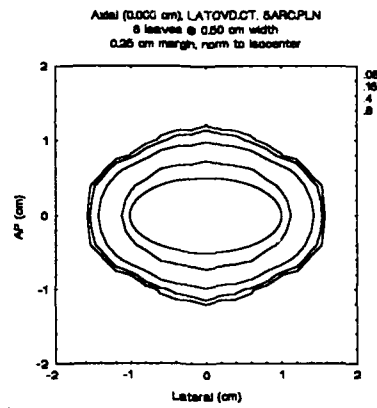


(b)

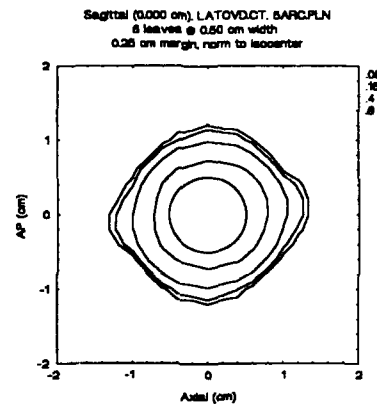


(c)

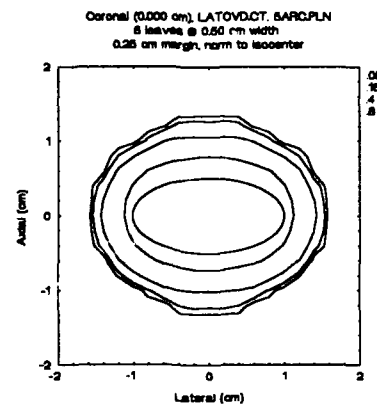
Figure 8-17: Lateral ovoid, 2.5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

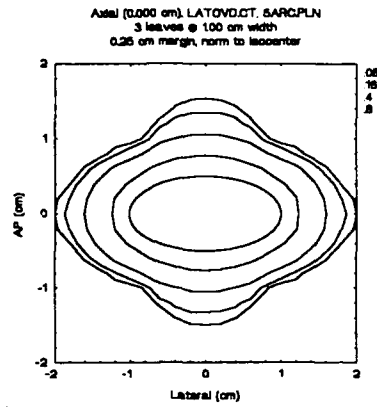


(b)

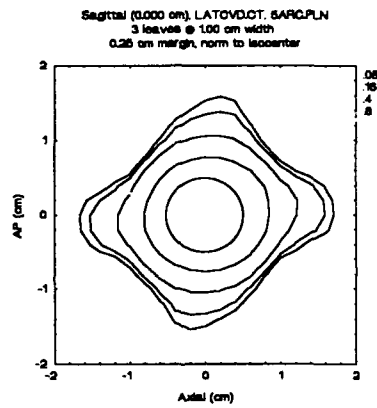


(c)

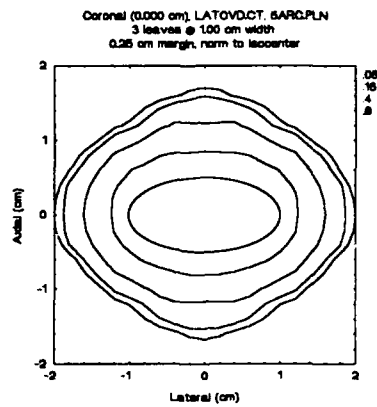
Figure 8-18: Lateral ovoid, 5 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)



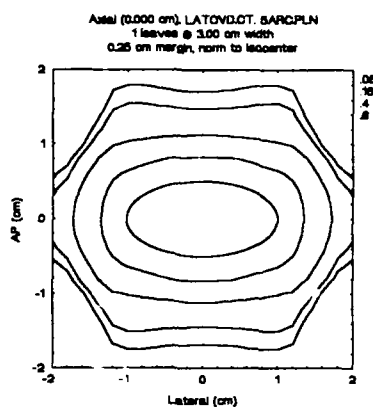
(b)



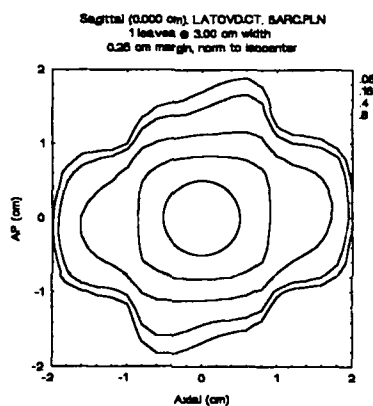
(c)

Figure 8-19: Lateral ovoid, 10 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut

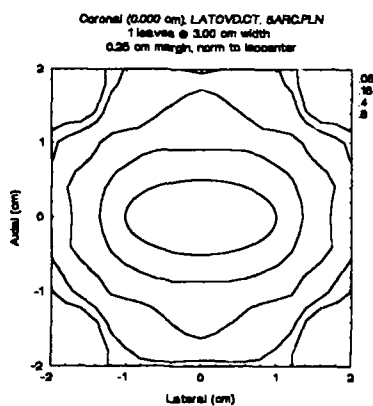




(a)

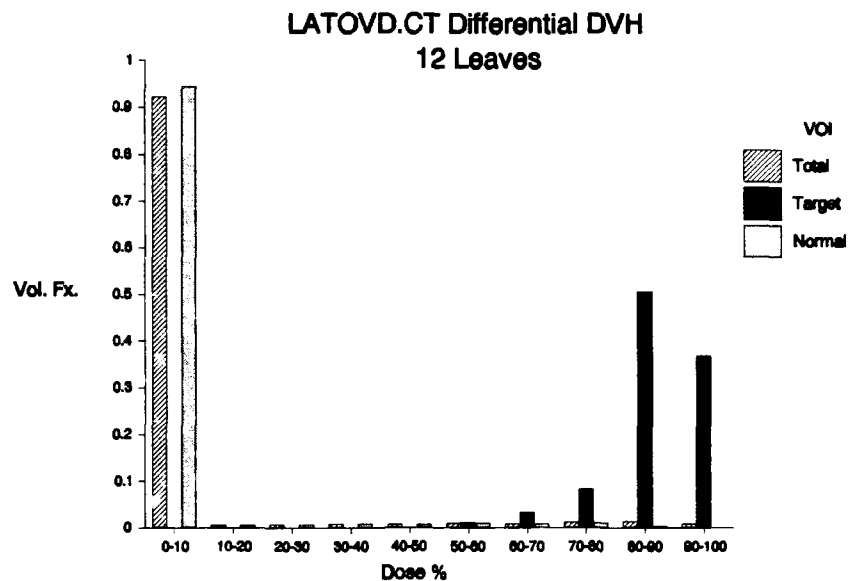


(b)

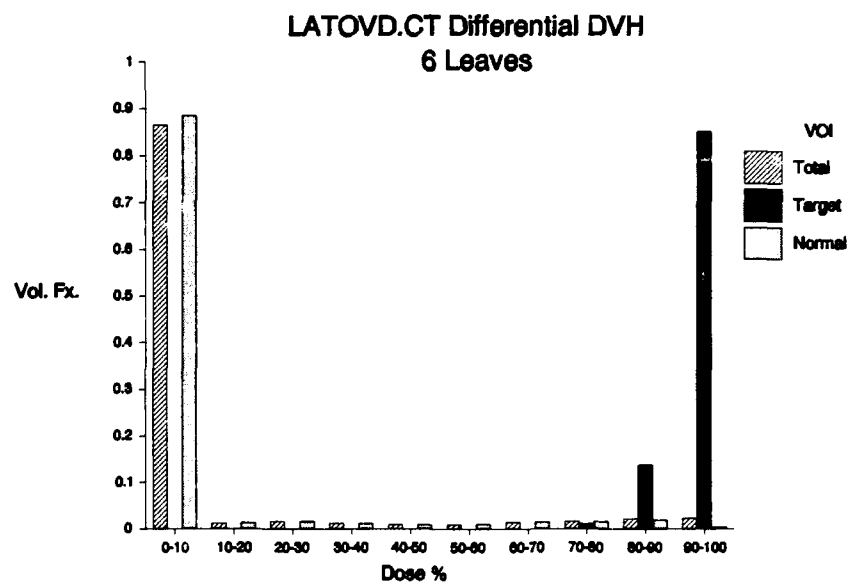


(c)

Figure 8-20: Lateral ovoid, 30 mm leaf localization  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

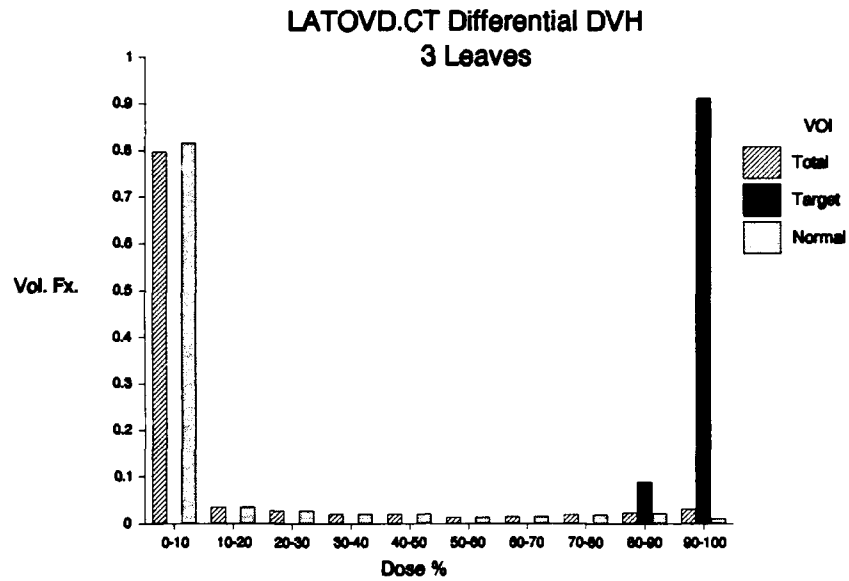


(a)

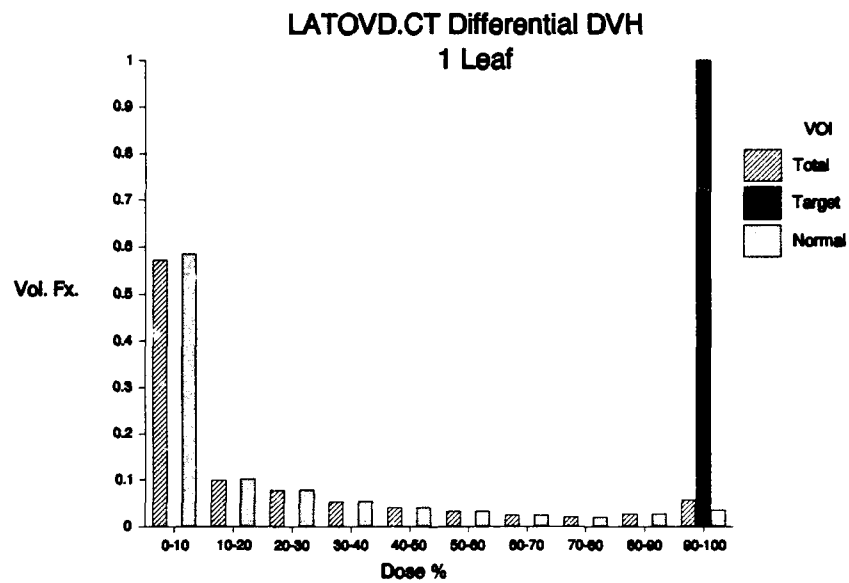


(b)

Figure 8-21: Lateral ovoid dose volume histogram  
(a) 2.5 mm leaf localization; (b) 5 mm leaf localization

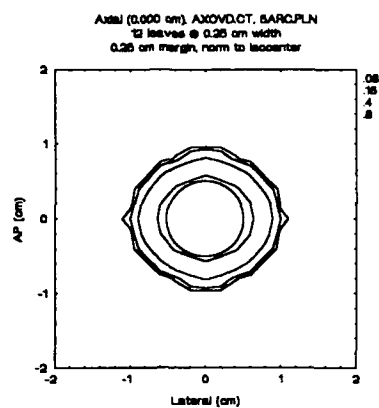


(c)

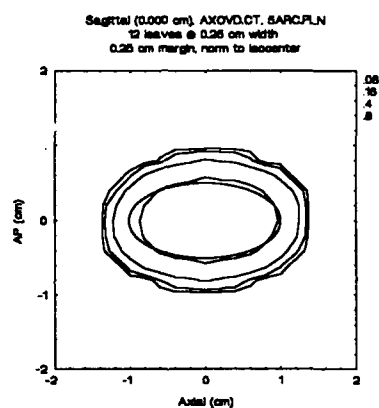


(d)

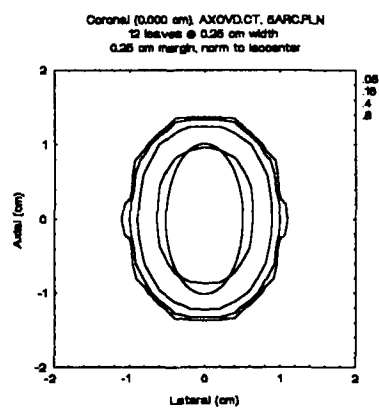
Figure 8-21 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization



(a)

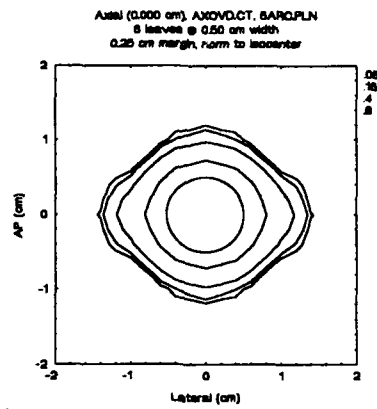


(b)

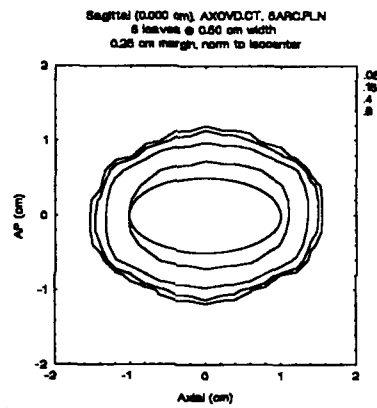


(c)

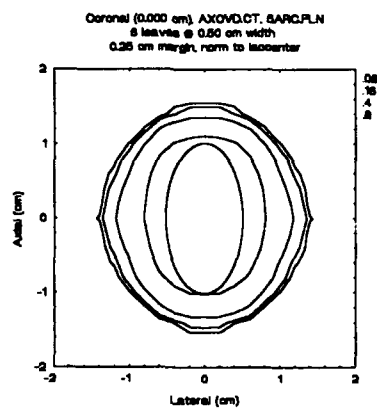
Figure 8-22: Axial ovoid, 2.5 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

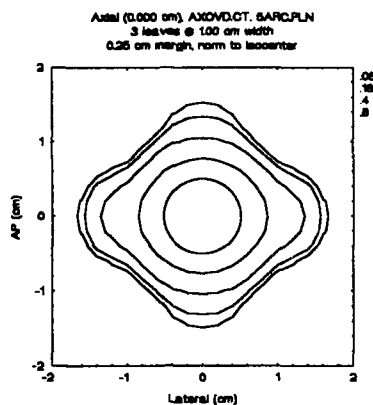


(b)

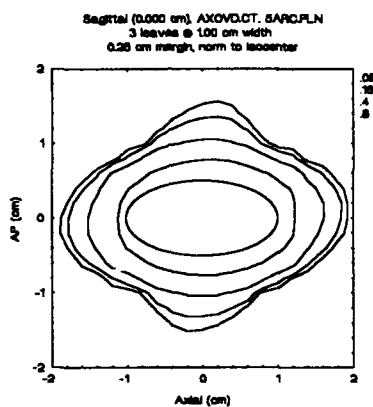


(c)

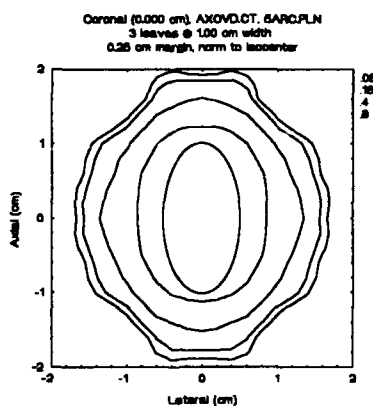
Figure 8-23: Axial ovoid, 5 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

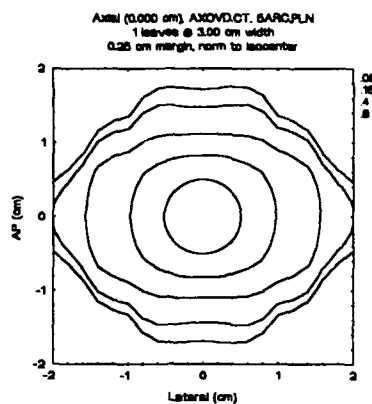


(b)

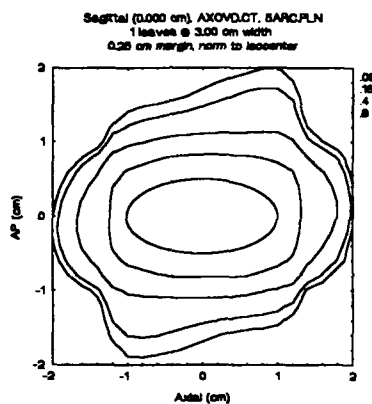


(c)

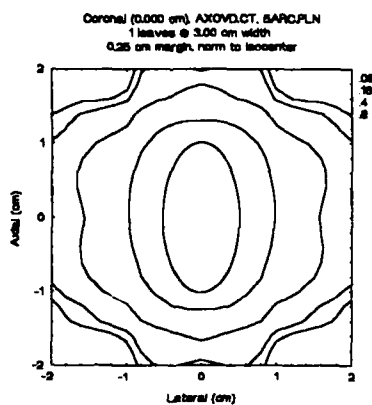
Figure 8-24: Axial ovoid, 10 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

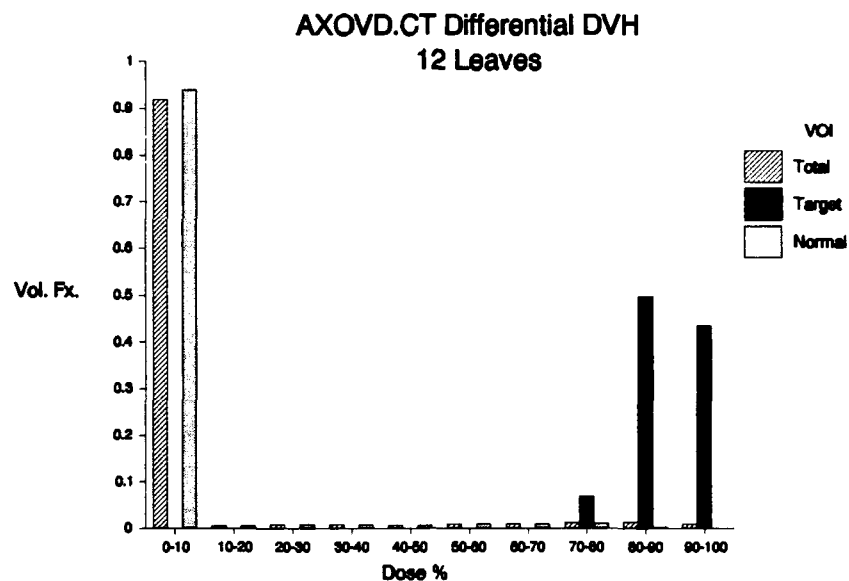


(b)

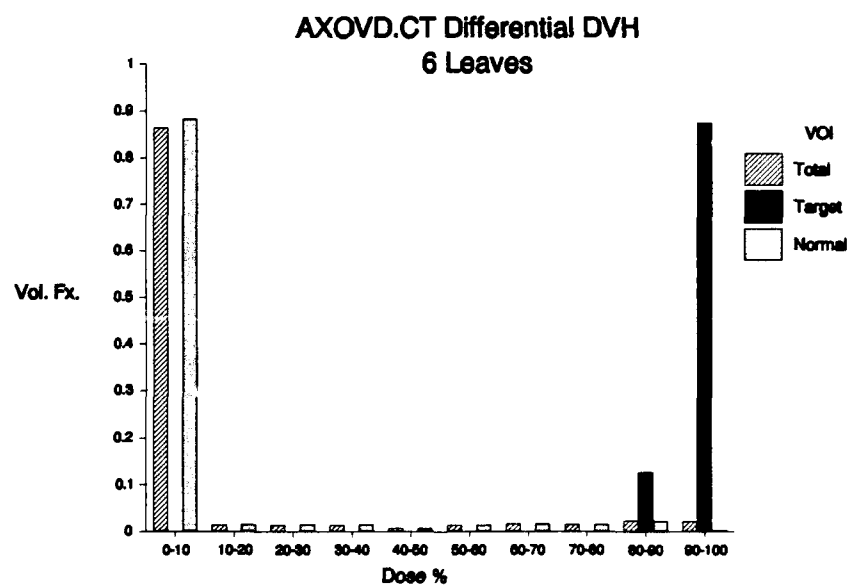


(c)

Figure 8-25: Axial ovoid, 30 mm leaf localization  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



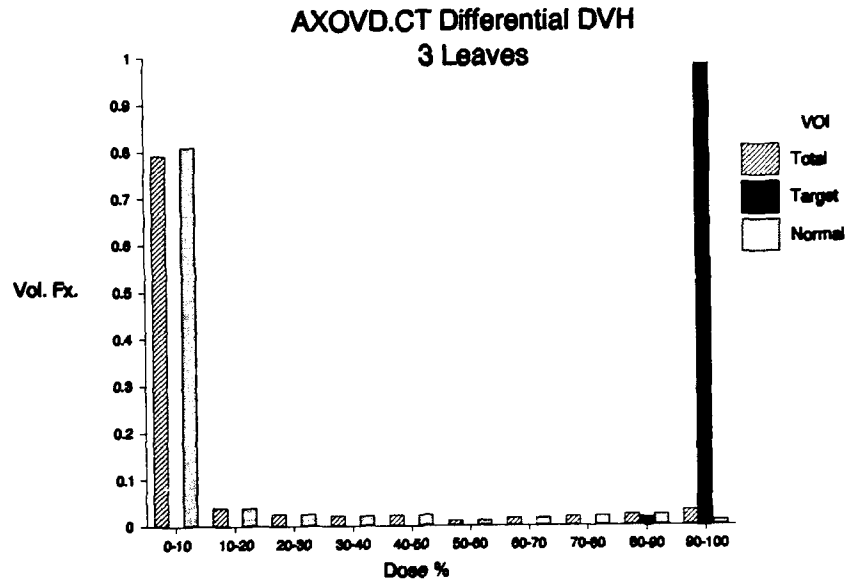
(a)



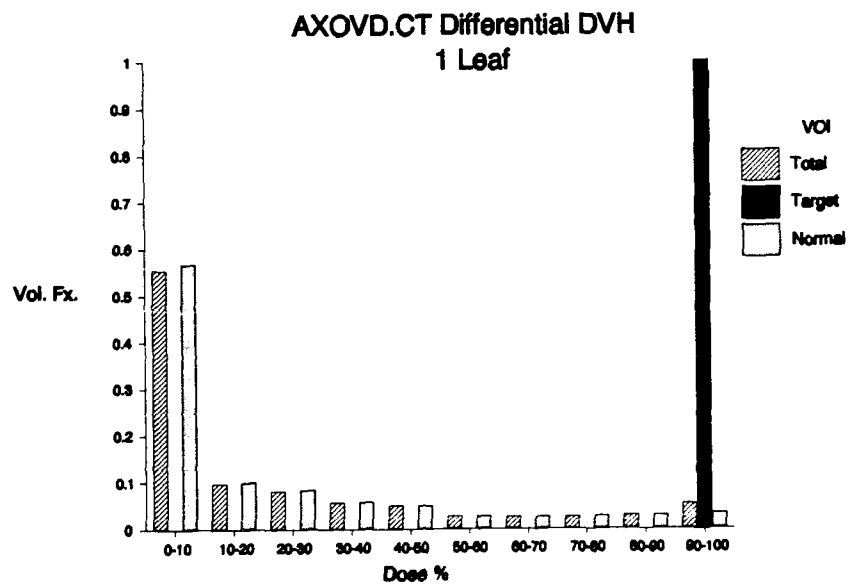
(b)

Figure 8-26: Axial ovoid dose volume histogram  
 (a) 2.5 mm leaf localization; (b) 5 mm leaf localization



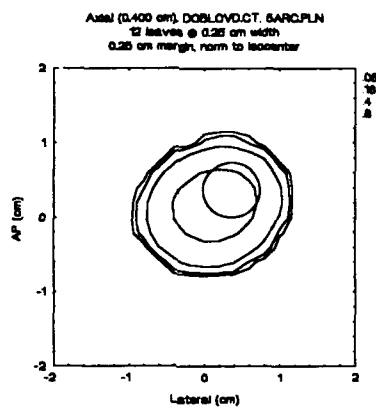


(c)

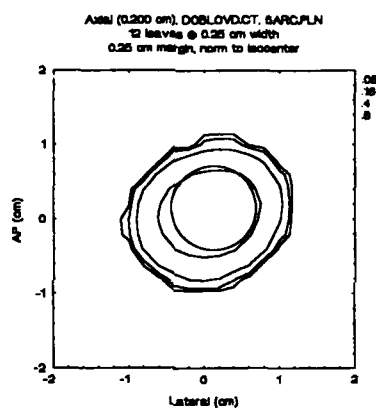


(d)

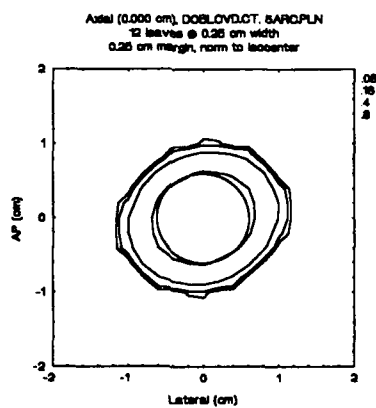
Figure 8-26 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization



(a)

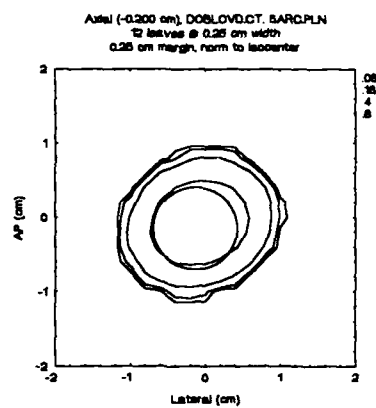


(b)

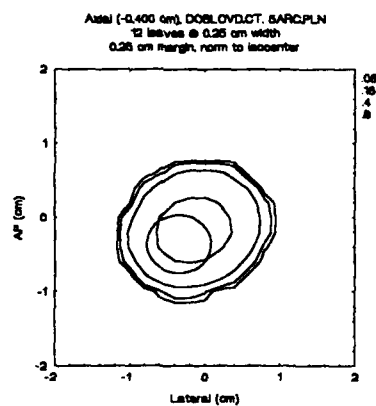


(c)

Figure 8-27: Double oblique ovoid, 2.5 mm leaf localization  
 (a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm

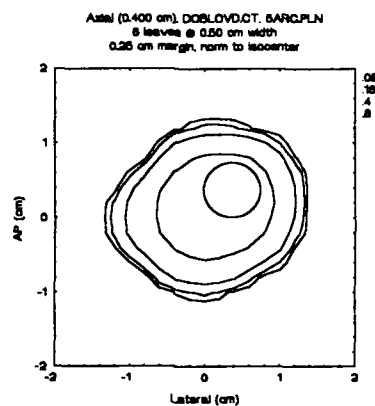


(d)

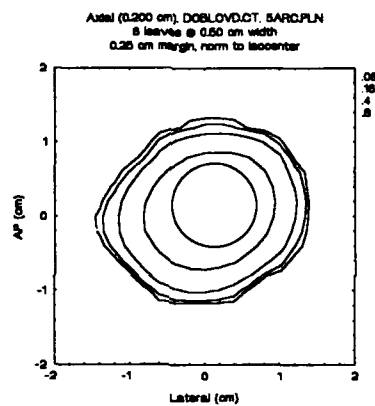


(e)

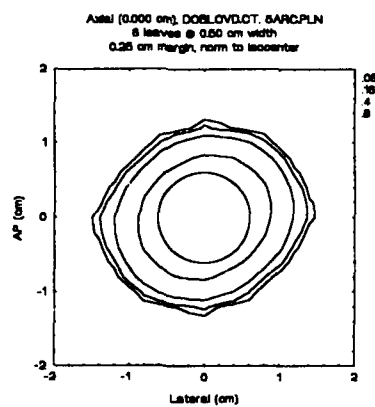
Figure 8-27 -- continued  
(d) Axial -2 mm; (e) Axial -4 mm



(a)

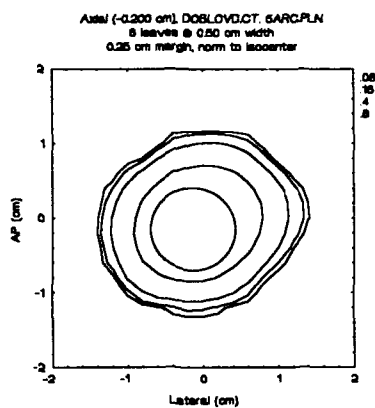


(b)

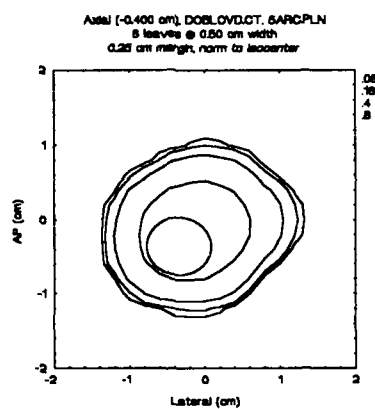


(c)

Figure 8-28: Double oblique ovoid, 5 mm leaf localization  
 (a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm

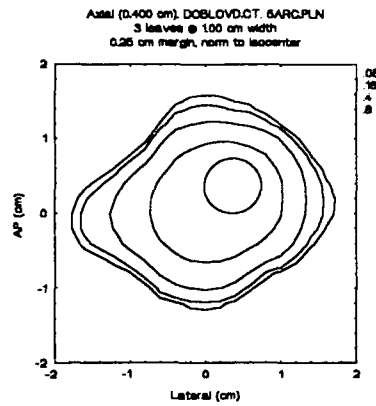


(d)

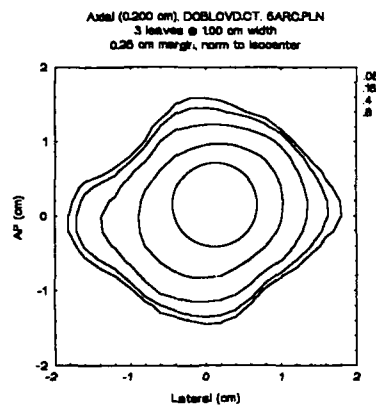


(e)

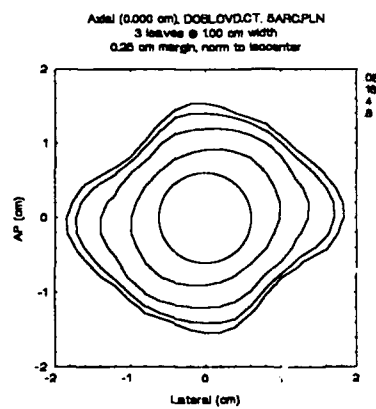
Figure 8-28 -- continued  
(d) Axial -2 mm; (e) Axial -4 mm



(a)

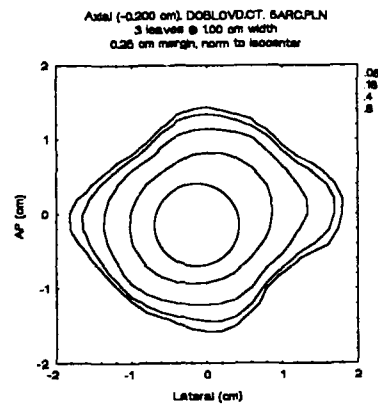


(b)

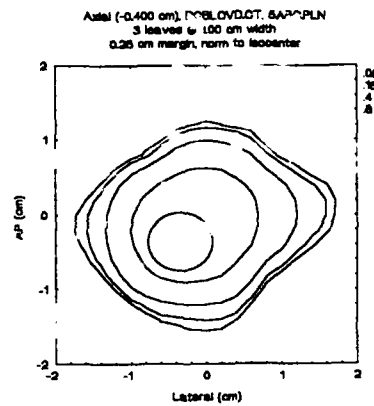


(c)

Figure 8-29: Double oblique ovoid, 10 mm leaf localization  
 (a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm

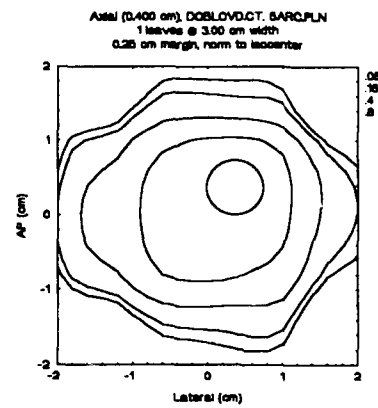


(d)

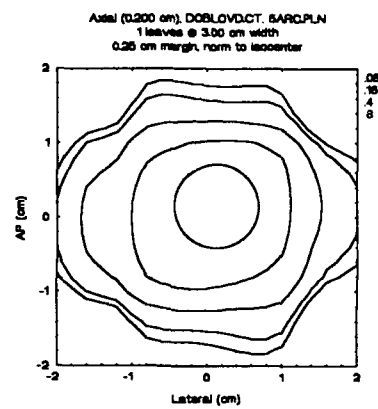


(e)

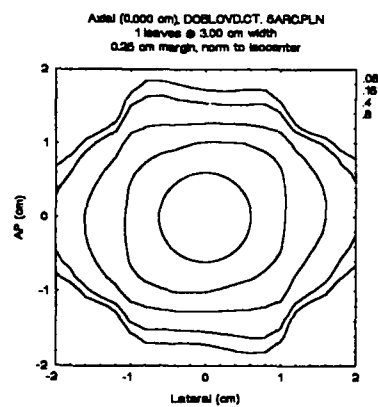
Figure 8-29 -- continued  
(d) Axial -2 mm; (e) Axial -4 mm



(a)



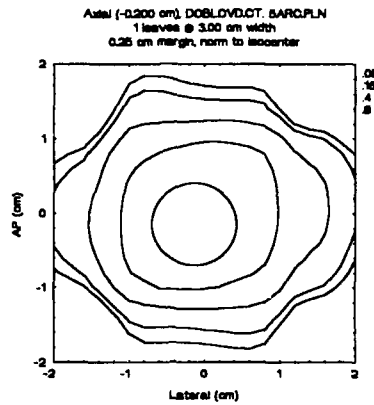
(b)



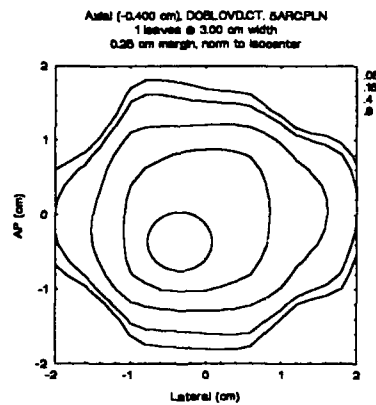
(c)

Figure 8-30: Double oblique ovoid, 30 mm leaf localization  
(a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm



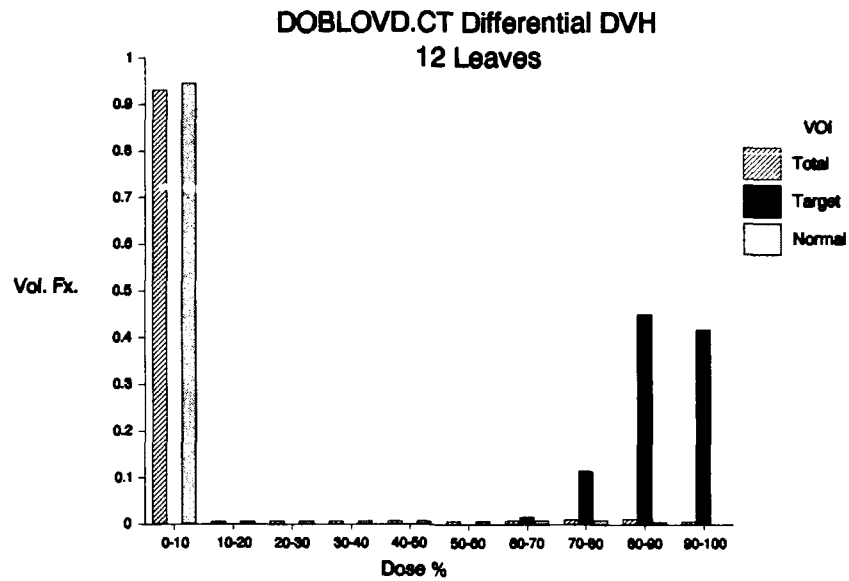


(d)

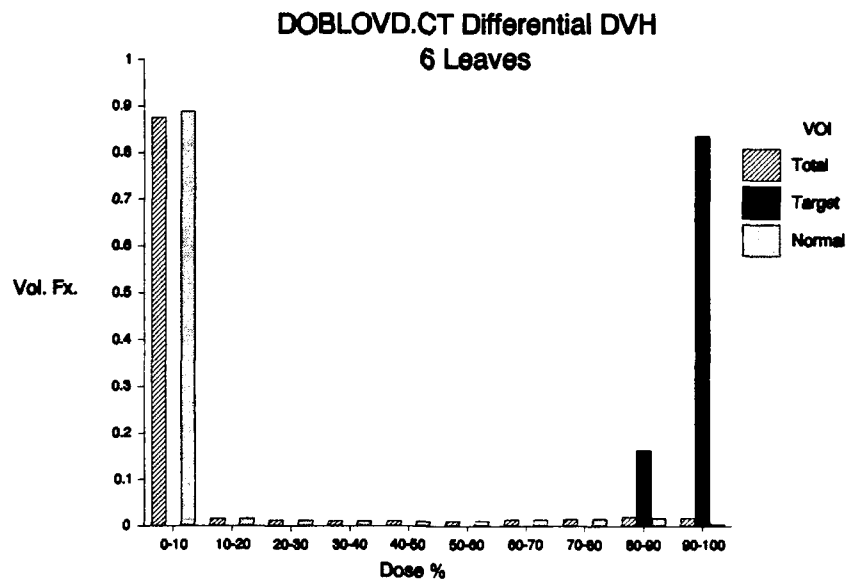


(e)

Figure 8-30 -- continued  
(d) Axial -2 mm; (e) Axial -4 mm

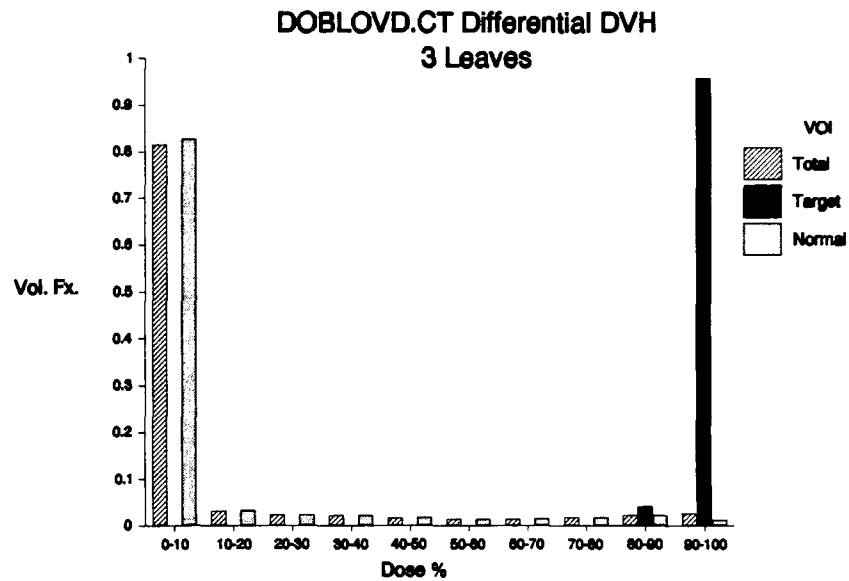


(a)

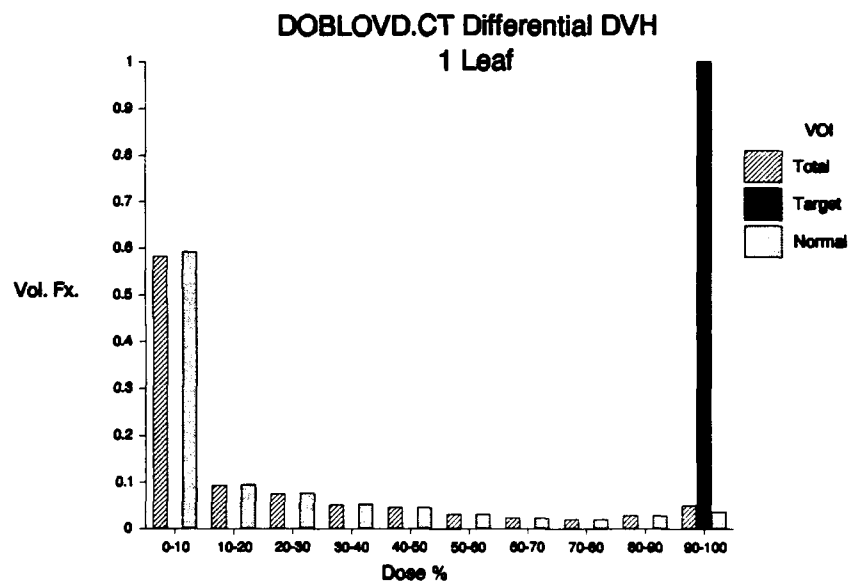


(b)

Figure 8-31: Double oblique ovoid dose volume histogram  
(a) 2.5 mm leaf localization; (b) 5 mm leaf localization



(c)



(d)

Figure 8-31 -- continued  
(c) 10 mm leaf localization; (d) 30 mm leaf localization

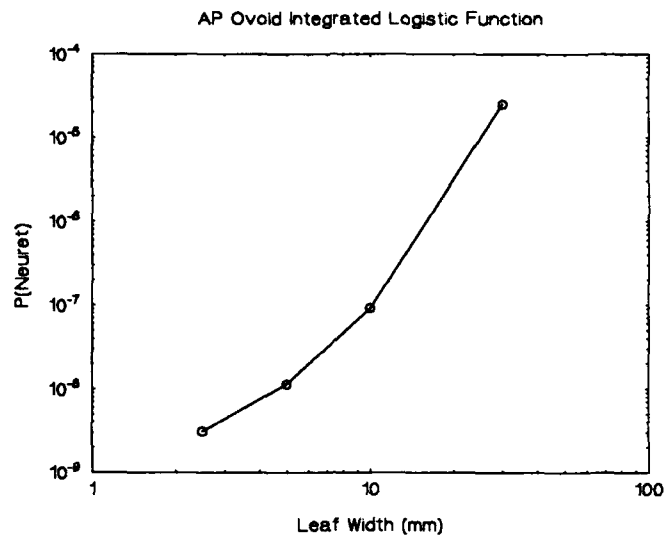


Figure 8-32: AP ovoid integrated logistic function

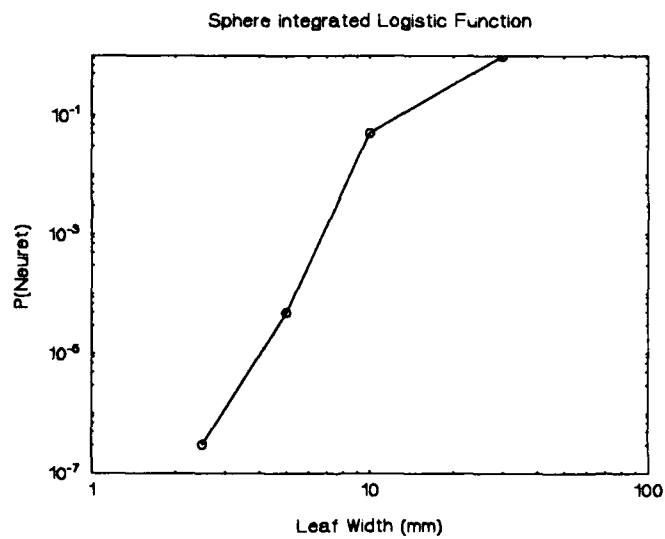


Figure 8-33: Sphere integrated logistic function

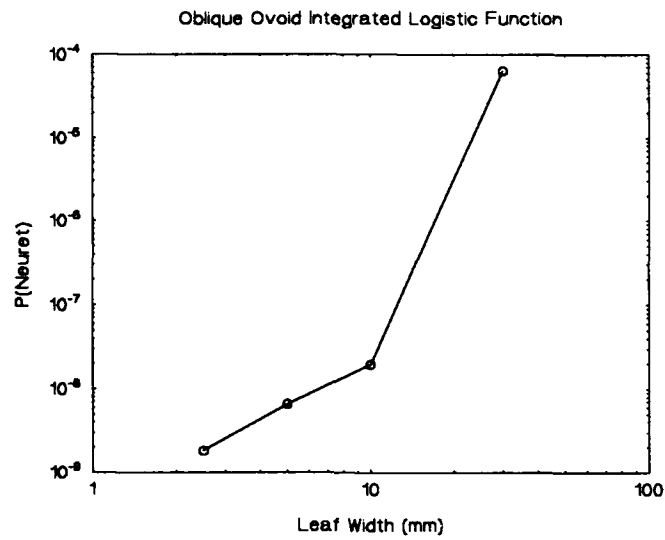


Figure 8-34: Oblique ovoid integrated logistic function

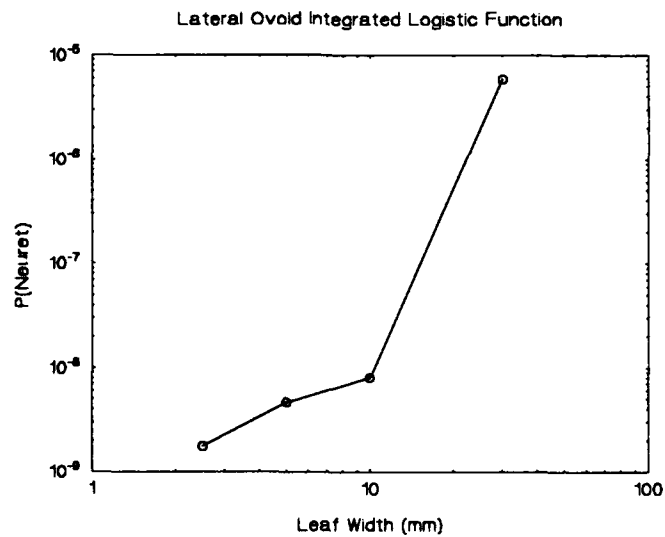


Figure 8-35: Lateral ovoid integrated logistic function

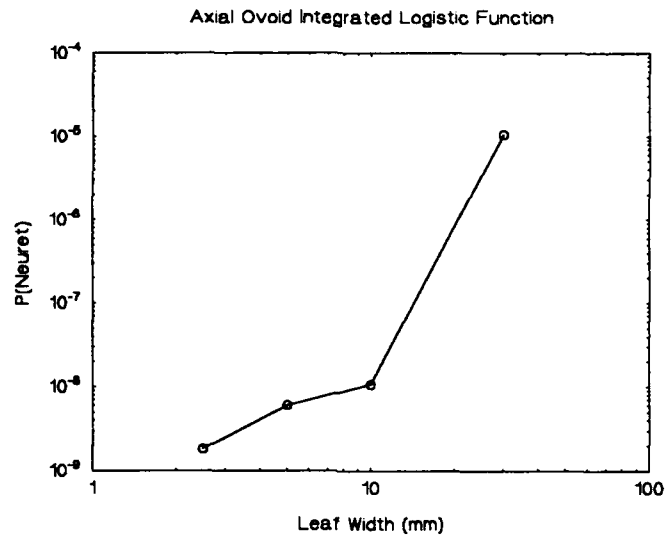


Figure 8-36: Axial ovoid integrated logistic function

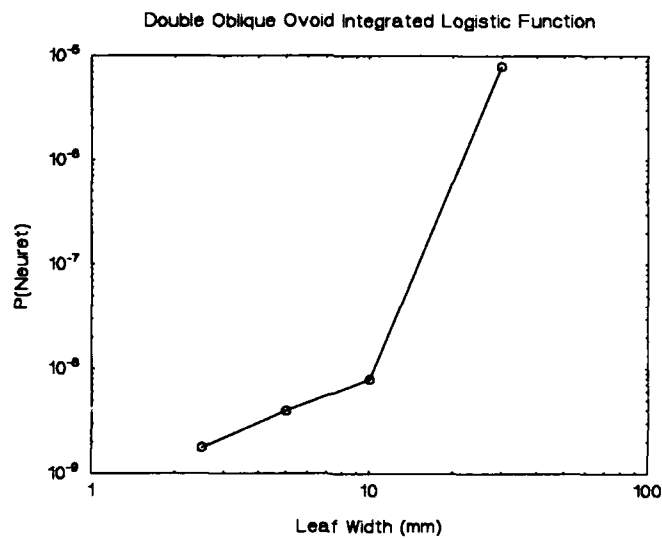
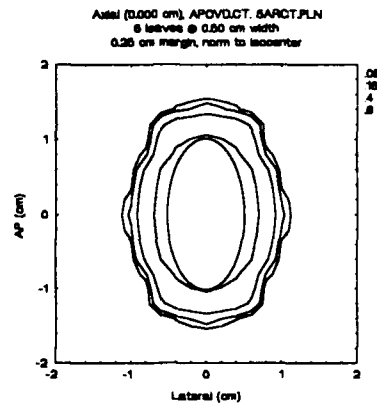
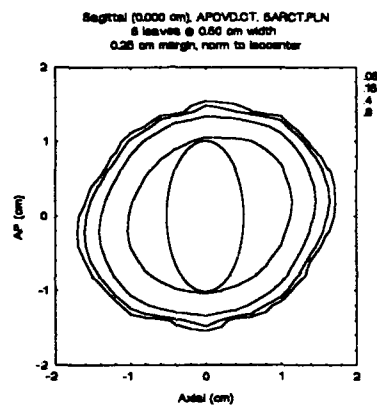


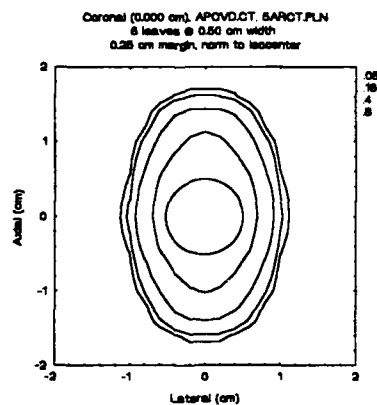
Figure 8-37: Double oblique ovoid integrated logistic function



(a)

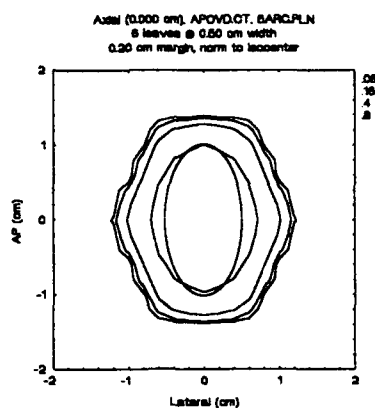


(b)

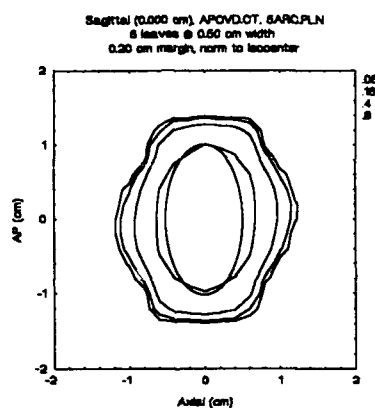


(c)

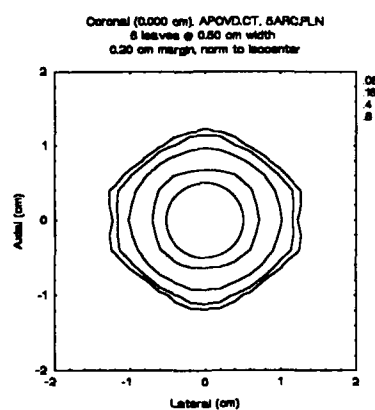
Figure 8-38: Effect of coronal arc compression, AP ovoid  
 (a) Axial plane; (b) Sagittal plane; (c) Coronal plane



(a)



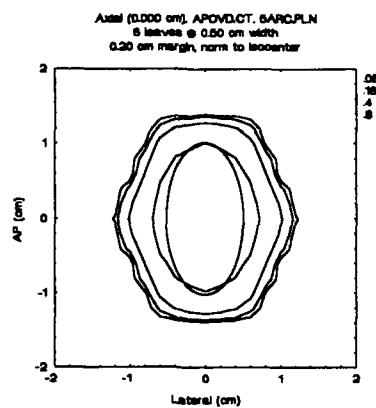
(b)



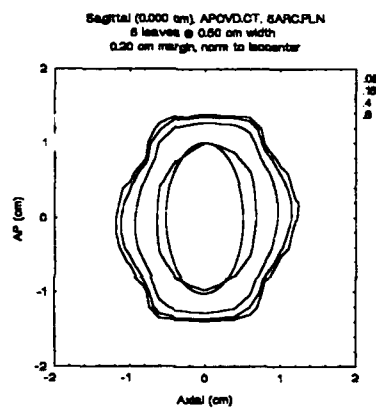
(c)

Figure 8-39: 5° gantry increment, AP ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

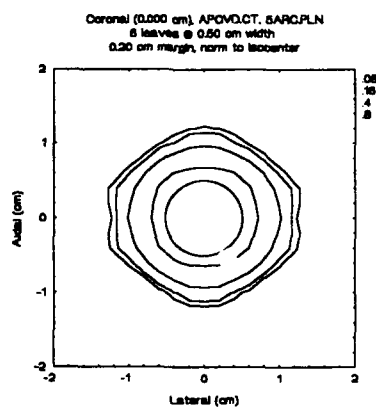




(a)

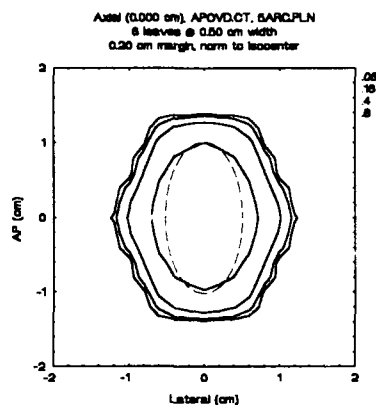


(b)

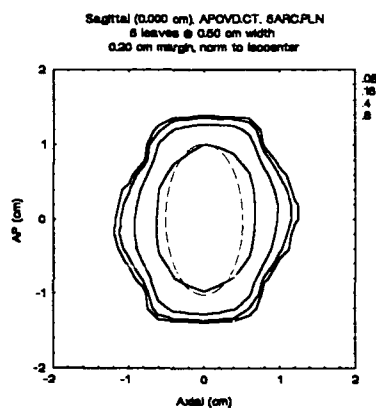


(c)

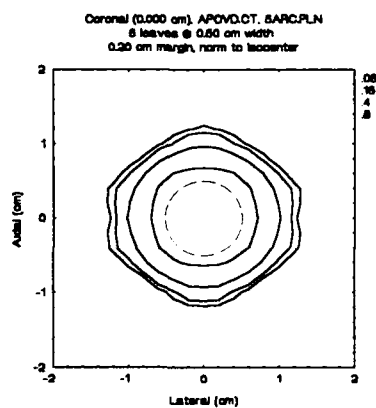
Figure 8-40: 10° gantry increment, AP ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

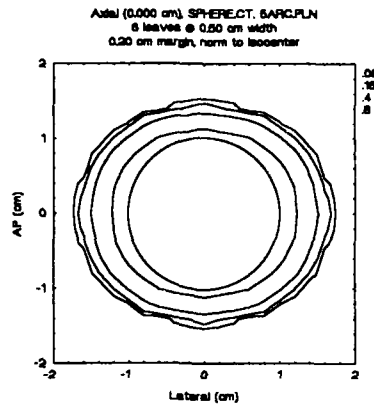


(b)

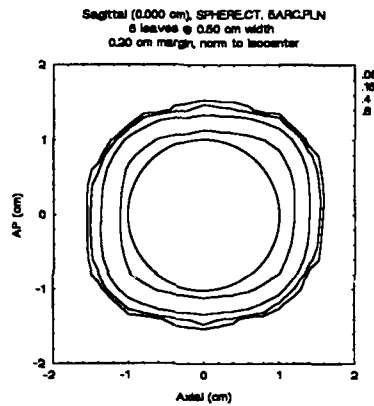


(c)

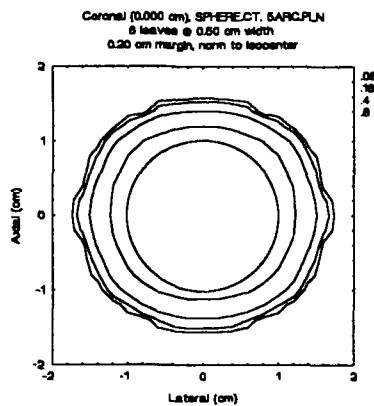
Figure 8-41: 20° gantry increment, AP ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

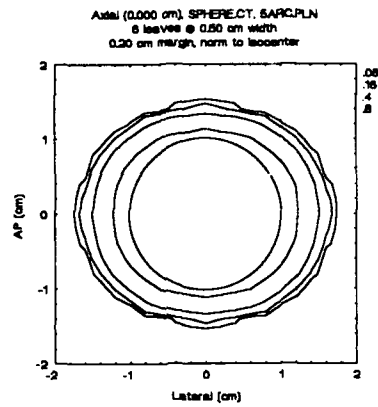


(b)

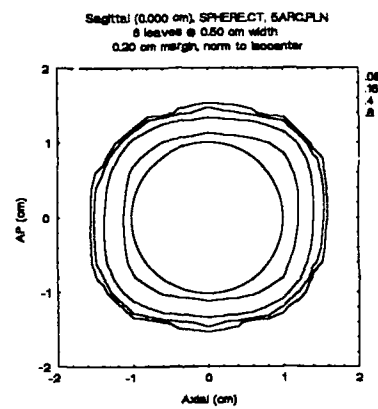


(c)

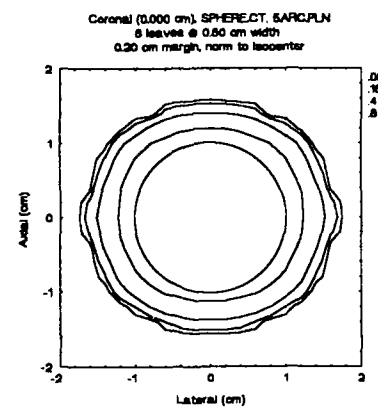
Figure 8-42: 5° gantry increment, sphere  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

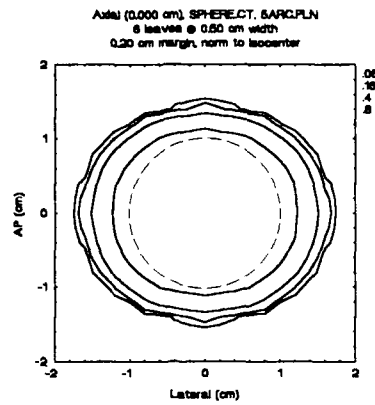


(b)

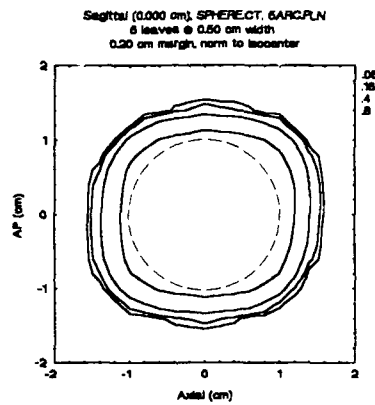


(c)

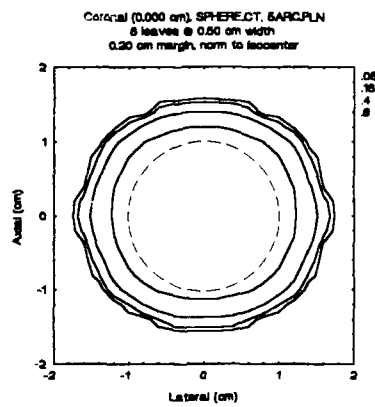
Figure 8-43: 10° gantry increment, sphere  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

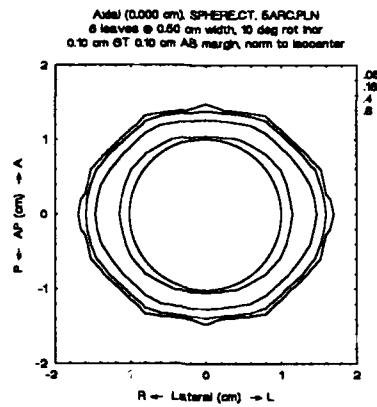


(b)

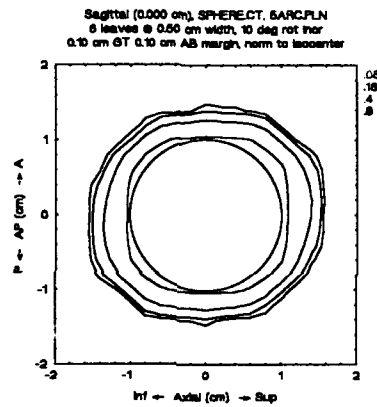


(c)

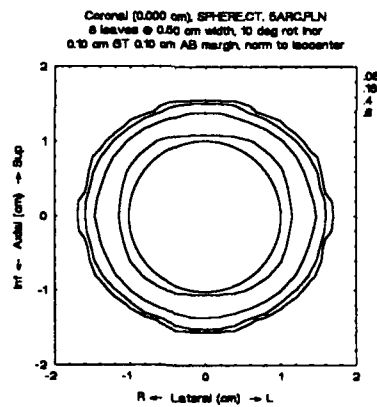
Figure 8-44: 20° gantry increment, sphere  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

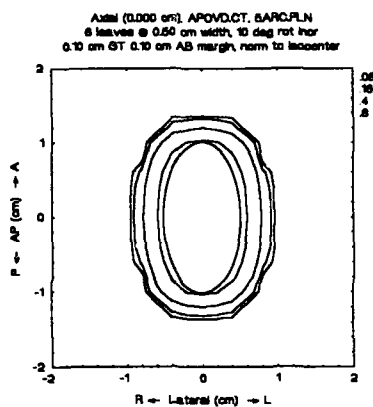


(b)

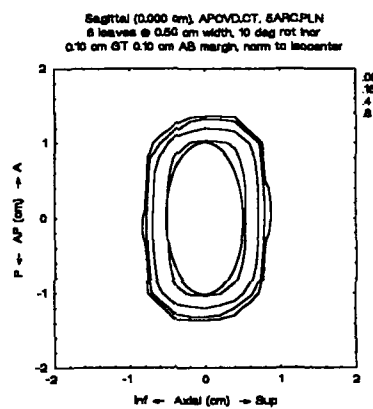


(c)

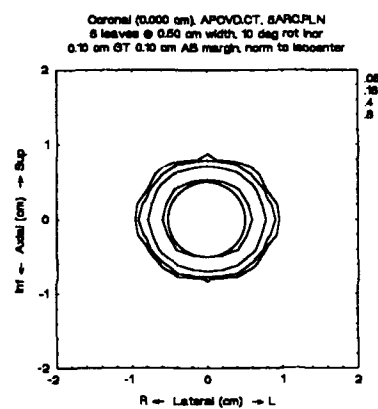
Figure 8-45: Empirical best fit, sphere  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

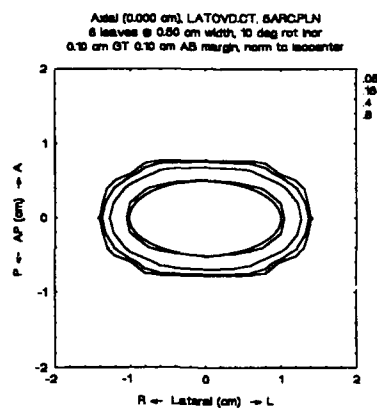


(b)

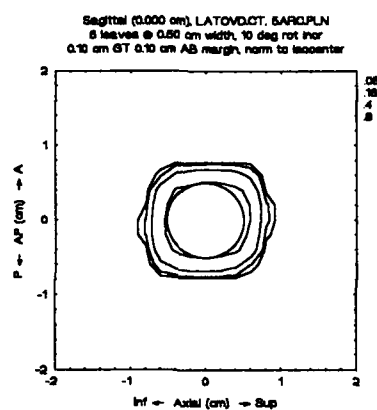


(c)

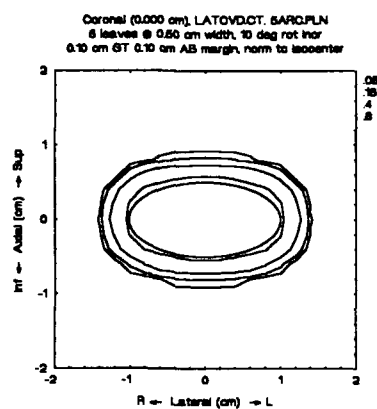
Figure 8-46: Empirical best fit, AP ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)



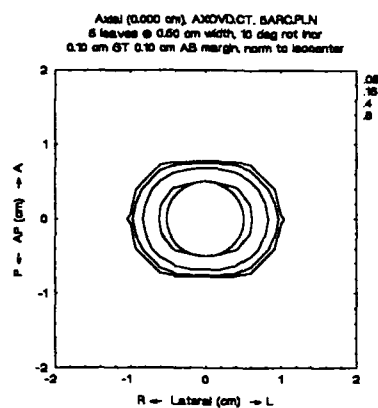
(b)



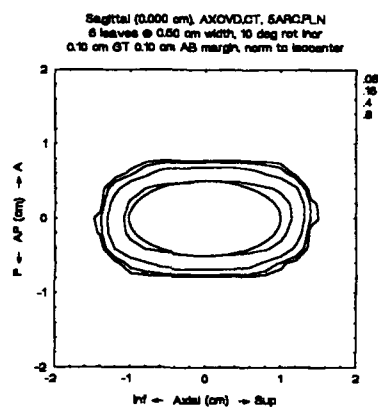
(c)

Figure 8-47: Empirical best fit, lateral ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

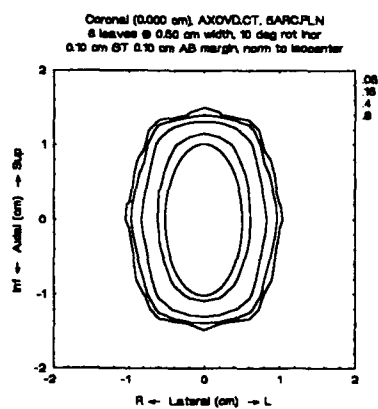




(a)

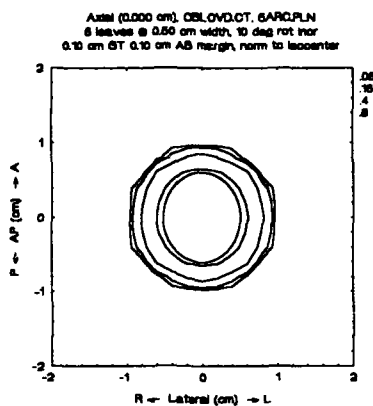


(b)

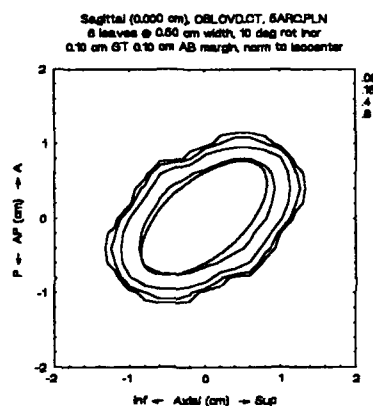


(c)

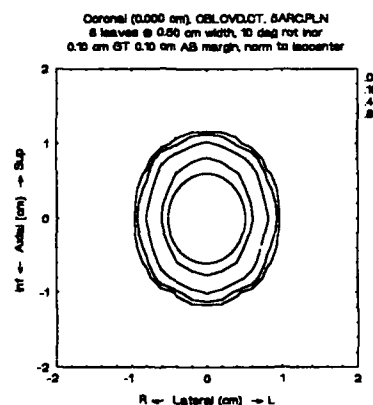
Figure 8-48: Empirical best fit, axial ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

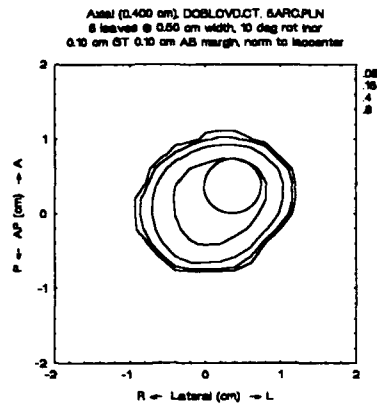


(b)

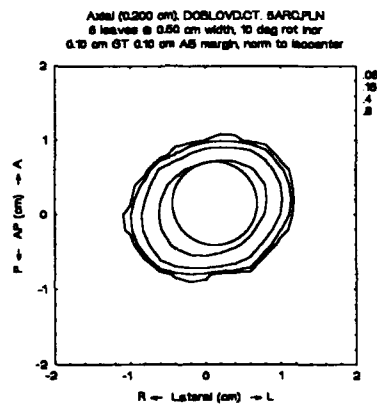


(c)

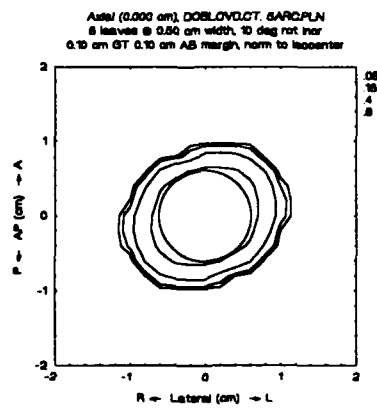
Figure 8-49: Empirical best fit, oblique ovoid  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

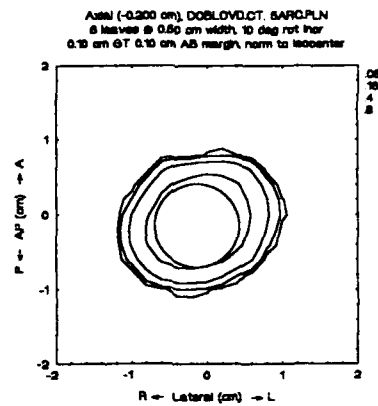


(b)

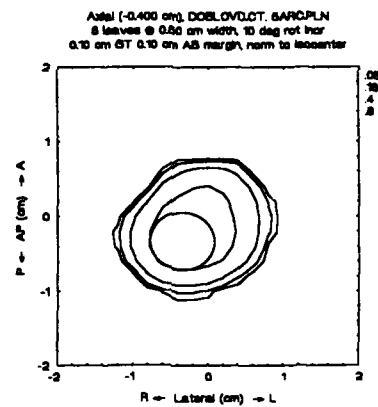


(c)

Figure 8-50: Empirical best fit, double oblique ovoid  
 (a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm

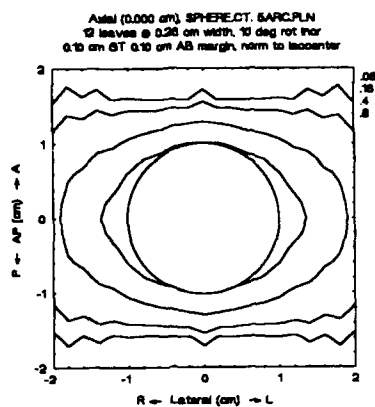


(d)

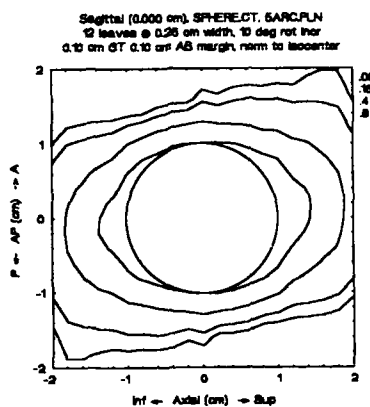


(e)

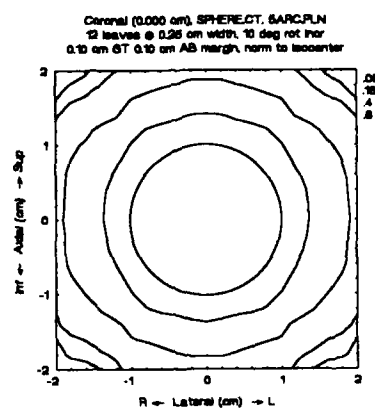
Figure 8-50 -- continued  
 (d) Axial -2 mm; (e) Axial -4 mm



(a)

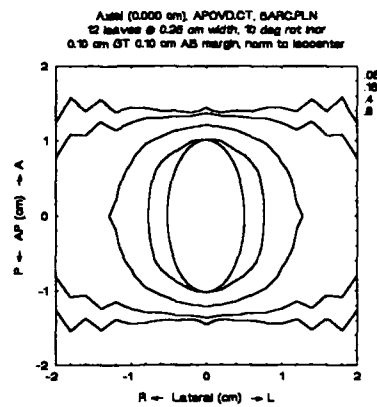


(b)

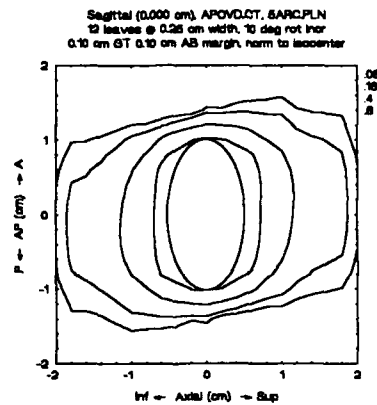


(c)

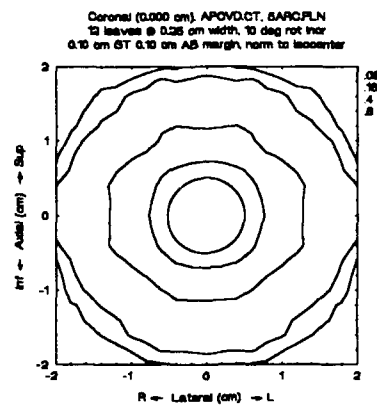
Figure 8-51: 2 jaw localization, sphere  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

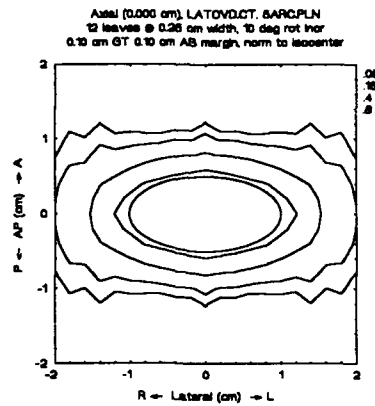


(b)

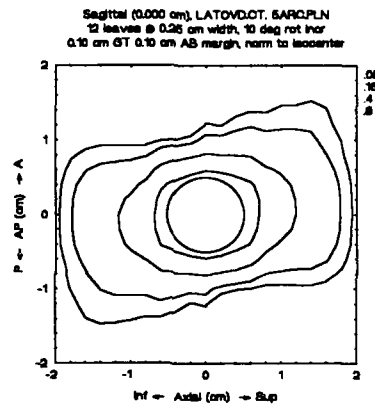


(c)

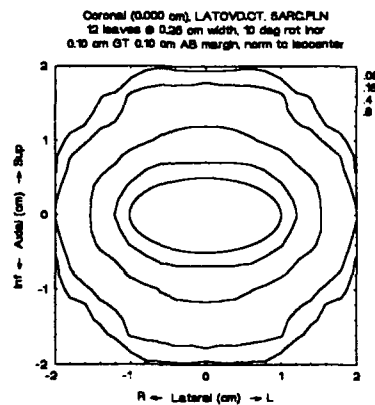
Figure 8-52: 2 jaw localization, AP ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

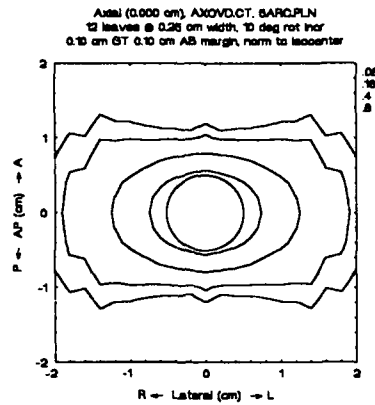


(b)

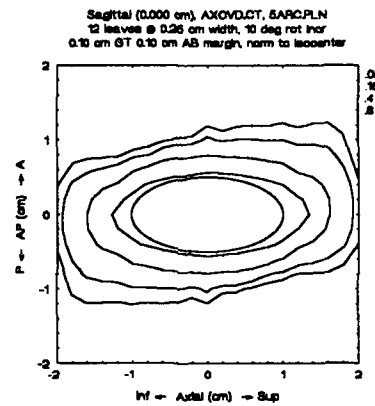


(c)

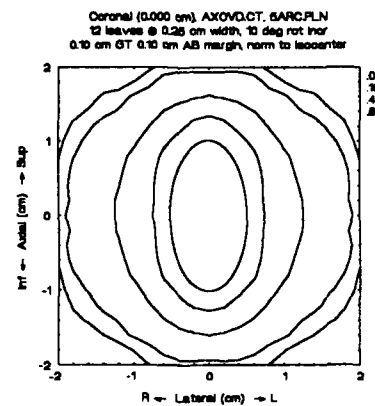
Figure 8-53: 2 jaw localization, lateral ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)



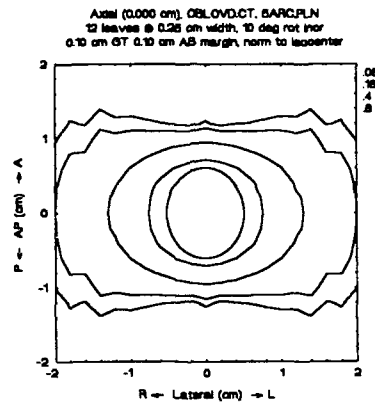
(b)



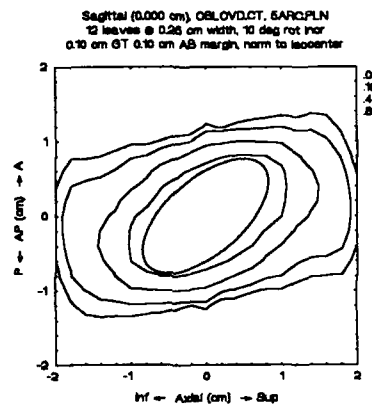
(c)

Figure 8-54: 2 jaw localization, axial ovoid  
 (a) Axial cut; (b) Sagittal cut; (c) Coronal cut

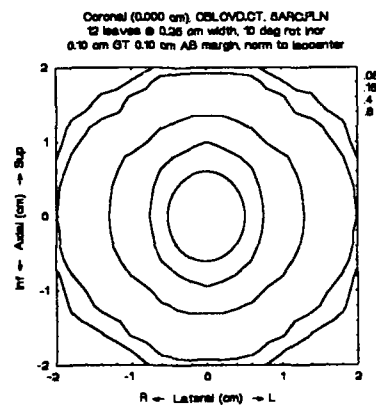




(a)

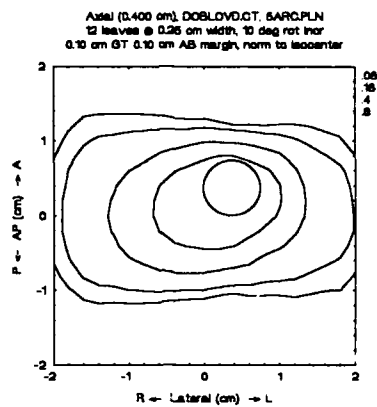


(b)

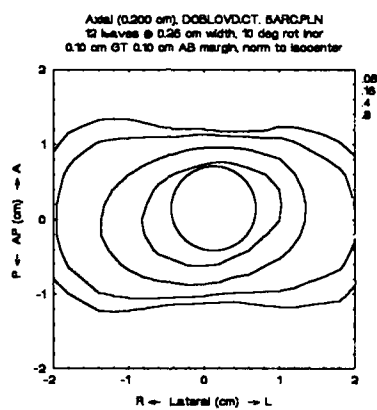


(c)

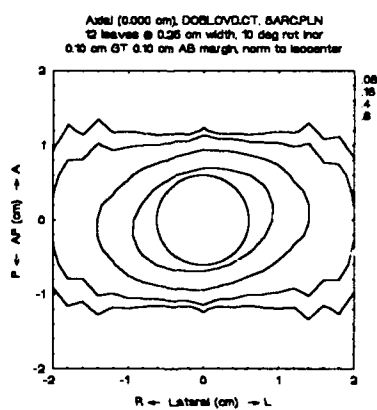
Figure 8-55: 2 jaw localization, oblique ovoid  
(a) Axial cut; (b) Sagittal cut; (c) Coronal cut



(a)

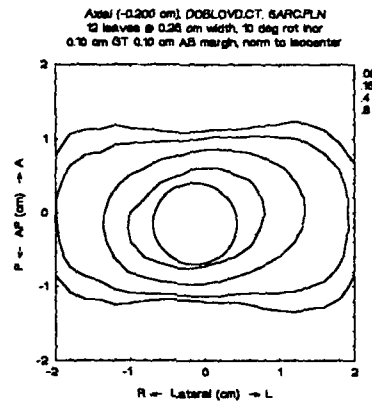


(b)

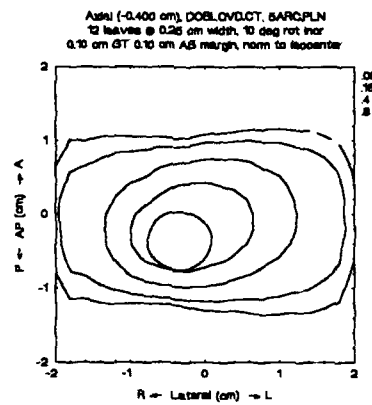


(c)

Figure 8-56: 2 jaw localization, double oblique ovoid  
 (a) Axial +4 mm; (b) Axial +2 mm; (c) Axial 0 mm



(d)



(e)

Figure 8-56 -- continued  
 (d) Axial -2 mm; (e) Axial -4 mm

## CHAPTER 9 CLINICAL EXAMPLES

Here we present clinical examples of previously treated cases and their resulting dosimetry using the standard circular collimators. The circular collimator dosimetry is compared with that resulting from a rotating collimator model and from both two jaw and four jaw conformal collimation as developed previously in this work.

Four targets were chosen to cover range of expected situations; a single isocenter small target, a single isocenter large target, a double isocenter extended target, and a triple isocenter irregular target. Targets were defined on all axial planes in which present and the resulting data was used as input for the localization program developed in chapter 4. The targets were localized using two jaws (from the A and B sides of the collimator) with a leaf width of 2.5 mm and using jaws from all four sides with a leaf width of 5 mm. For the rotating collimator model, the localization program was modified to allow manual collimator rotation followed by automatic localization of the input target with one leaf per collimator side, in the manner of Leavitt et al. [Lea91], resulting in the best fit rectangular col-

limation at each discrete gantry/table position. All localized targets use a standard nine arc treatment plan.

Dosimetry was produced at several axial positions, as well as were dose volume histograms. The dosimetry programs were modified by the addition of an extra rotation operation (to rotate the collimator) which then calculated a simple rectangular field, following the rectangular dosimetry of Suh [Suh90]. The dose volume histograms were further evaluated using Neuret weighting of the integrated logistic function (appendix E).

#### Case 1

Case 1, the small single isocenter target, was an arteriovenous malformation (AVM) in the right centrum. This target is illustrated in figure 9-1. The original treatment plan prescribes nine arcs using a 14mm circular collimator (see table 3-1). The resulting dosimetry is shown in figure 9-2. Figure 9-3 details the dosimetry of the rotating collimator, figure 9-4 that of the two jaw conformal collimation, and figure 9-5 that of the four jaw conformal collimation. All fit the target to the 80% line, except for the first circular slice, our outline of which probably included a vein not treated originally.

The integrated logistic function operation on the dose volume histograms is detailed in table 9-1, with the histograms shown in figure 9-6. The target volume results, in which a higher value implies a higher integrated dose, we

see that all values are similar, indicating that all dosimetry modes cover the target equally well. The normal volume results indicate the integrated dose over the volume exclusive of the target, where a lower number implies greater normal tissue sparing.

Table 9-1: Case 1 integrated logistic function

Mode	Target tissue	Normal tissue
Circular coll.	0.398	$5.58 \times 10^{-8}$
Rotating coll.	0.482	$3.15 \times 10^{-4}$
2 jaw conformal	0.465	$1.37 \times 10^{-9}$
4 jaw conformal	0.429	$1.69 \times 10^{-9}$

The rotating collimator is seen in the dosimetry to spread on the lower isodose lines, leading to a relatively high value, however the rest of the results are comparable. In this case, any of the delivery modes could be used with similar results predicted.

#### Case 2

Case 2, the large single isocenter target, was an AVM in the right frontal lobe. The target is shown in figure 9-7. It was originally treated with the nine standard arcs using a 30 mm collimator (see table 3-1). The resulting dosimetry is shown in figure 9-8. Figure 9-9 details the dosimetry of the rotating collimator, figure 9-10 that of the two jaw conformal collimation, and figure 9-11 that of the four jaw conformal collimation. All fit the target well

to the 80% line, with the exception of the first circularly collimated slice, also most likely a vein.

The integrated logistic function results are shown in table 9-2, from the dose volume histograms in figure 9-12. All targets are equally well covered, with the best normal tissue sparing given by the conformal collimation modes.

Table 9-2: Case 2 integrated logistic function

Mode	Target tissue	Normal tissue
Circular coll.	0.999	$4.92 \times 10^{-1}$
Rotating coll.	0.999	$9.94 \times 10^{-1}$
2 jaw conformal	0.999	$7.76 \times 10^{-2}$
4 jaw conformal	0.998	$1.10 \times 10^{-8}$

### Case 3

Case 3, the double isocenter target, is shown in figure 9-13. This target was an AVM in the left occipital lobe. The prescribed treatment plan is given in table 9-3.

The resulting dosimetry is shown in figure 9-14 for the circular collimator. The other three modes used a standard nine arc set (see table 3-1) and the dosimetry is shown in figure 9-15 for the rotating collimator, figure 9-16 for the two jaw conformal collimation, and figure 9-17 for the four jaw conformal collimation. The latter three modes fit the target well to the 80% line. The prescribed treatment plan also fits reasonably well, given the difficulties of multiple isocenter planning and treatment.

Table 9-3: Case 3 treatment plan

Iso-center	Coll (mm)	Table angle	Gantry start	Gantry stop	Arc weight
1	20	10°	30°	130°	1
1	20	46°	30°	130°	1
1	20	82°	30°	130°	1
1	20	334°	230°	330°	1
1	20	298°	230°	330°	1
2	18	28°	30°	130°	1
2	18	64°	30°	130°	1
2	18	352°	230°	330°	1
2	18	316°	230°	330°	1
2	18	280°	230°	330°	1

The integrated logistic function results are shown in table 9-4, from the dose volume histograms illustrated in figure 9-18. All target volumes receive similar coverage while, again, the best normal tissue sparing is generated by the conformal modes.

Table 9-4: Case 3 integrated logistic function

Mode	Target tissue	Normal tissue
Circular coll.	0.948	$4.69 \times 10^{-1}$
Rotating coll.	0.988	$9.54 \times 10^{-1}$
2 jaw conformal	0.985	$1.22 \times 10^{-3}$
4 jaw conformal	0.979	$5.87 \times 10^{-9}$

#### Case 4

Case 4, the triple isocenter target, is also an AVM located in the left internal capsule. This target is il-



illustrated in figure 9-19. The treatment plan prescribed is given in table 9-5. The dosimetry resulting from this plan is shown in figure 9-21. Note that this plan, as with most multiple isocenter plans, is prescribed to the 70% line. This line is added to the figure for reference, and fits the target well.

Table 9-5: Case 4 treatment plan

Iso-center	Coll. (mm)	Table	Gantry start	Gantry stop	Arc weight
1	16	20°	30°	130°	1
1	16	55°	30°	130°	1
1	16	340°	230°	330°	1
1	16	305°	230°	330°	1
1	16	270°	230°	330°	1
2	16	20°	30°	130°	0.75
2	16	55°	30°	130°	0.75
2	16	340°	230°	330°	0.75
2	16	305°	230°	330°	0.75
2	16	270°	230°	330°	0.75
3	16	20°	30°	130°	0.75
3	16	55°	30°	130°	0.75
3	16	340°	230°	330°	0.75
3	16	305°	230°	330°	0.75
3	16	270°	230°	330°	0.75

The other three modes are, as before, given a standard nine arc plan (see table 3-1) with a dose prescription to the 80% line. The result of these plans are shown in figure 9-22 for the rotating collimator, figure 9-23 for the two

jaw conformal, and figure 9-24 for the four jaw conformal collimators. These dosimetries also provide a good target fit.

The integrated logistic function was evaluated on the dose volume histograms shown in figure 9-24. The result is seen in table 9-6.

Table 9-6: Case 4 integrated logistic function

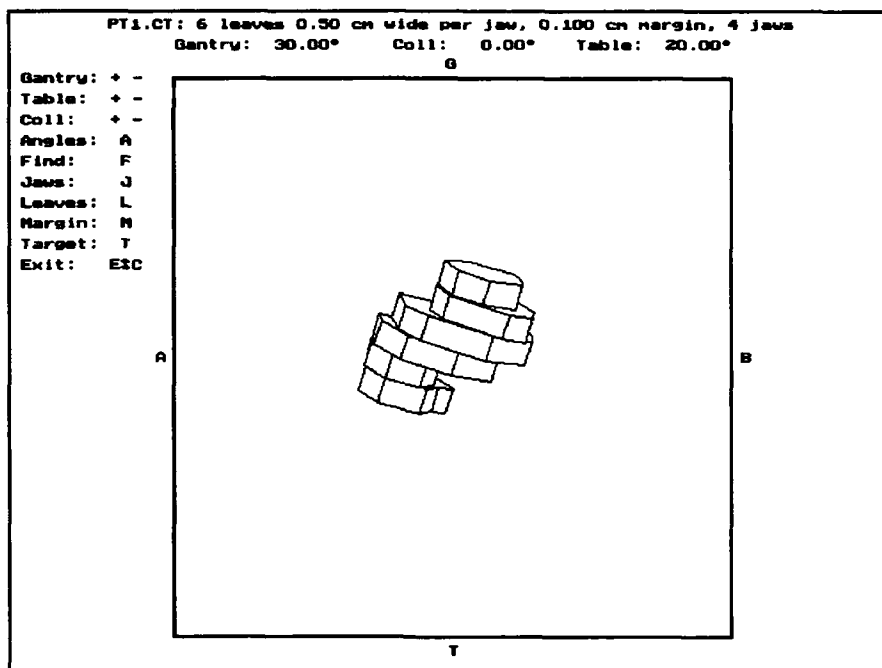
Mode	Target tissue	Normal tissue
Circular coll.	0.637	$4.61 \times 10^{-1}$
Rotating coll.	0.722	$3.76 \times 10^{-2}$
2 jaw conformal	0.710	$2.49 \times 10^{-7}$
4 jaw conformal	0.665	$1.37 \times 10^{-9}$

The target is, again, covered similarly in all modes, while the conformal modes provide the best normal tissue sparing.

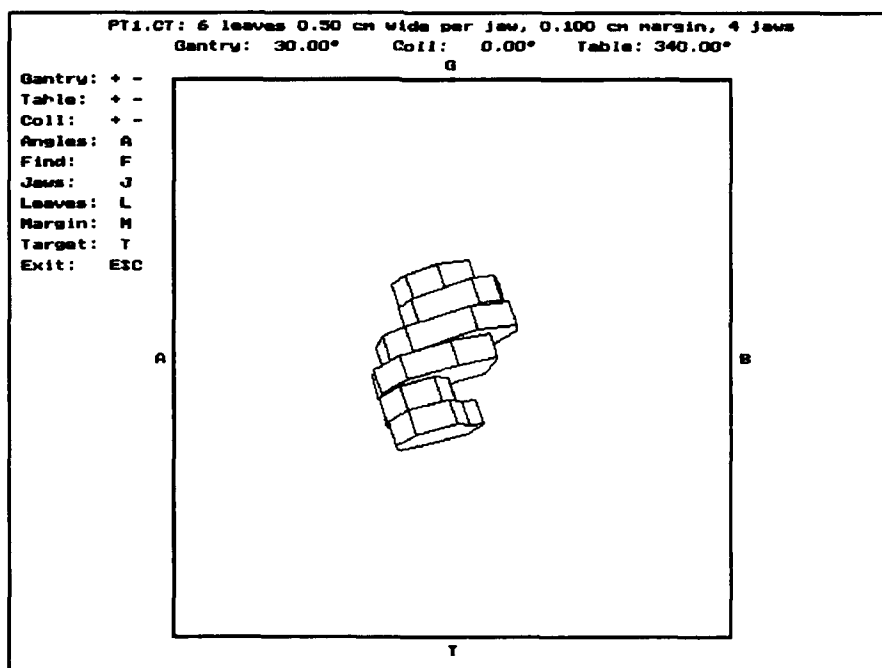
#### Conclusion

Four treatment delivery modalities have been compared on each of four representative targets. Dosimetry has been prepared on each and the dosimetry has been analyzed with the integrated logistic function. In each case, good target coverage and minimal normal tissue dose has been demonstrated using the conformal modes in preference to either the standard circular or the rotating collimator modes. In particular contrast to the multiple isocenter circular collimator mode, the conformal modes provide a demonstrated

homogeneous target coverage that may be of advantage in reducing complications of treatment.



(a)



(b)

Figure 9-1: Case 1 target  
 (a) View 1; (b) View 2

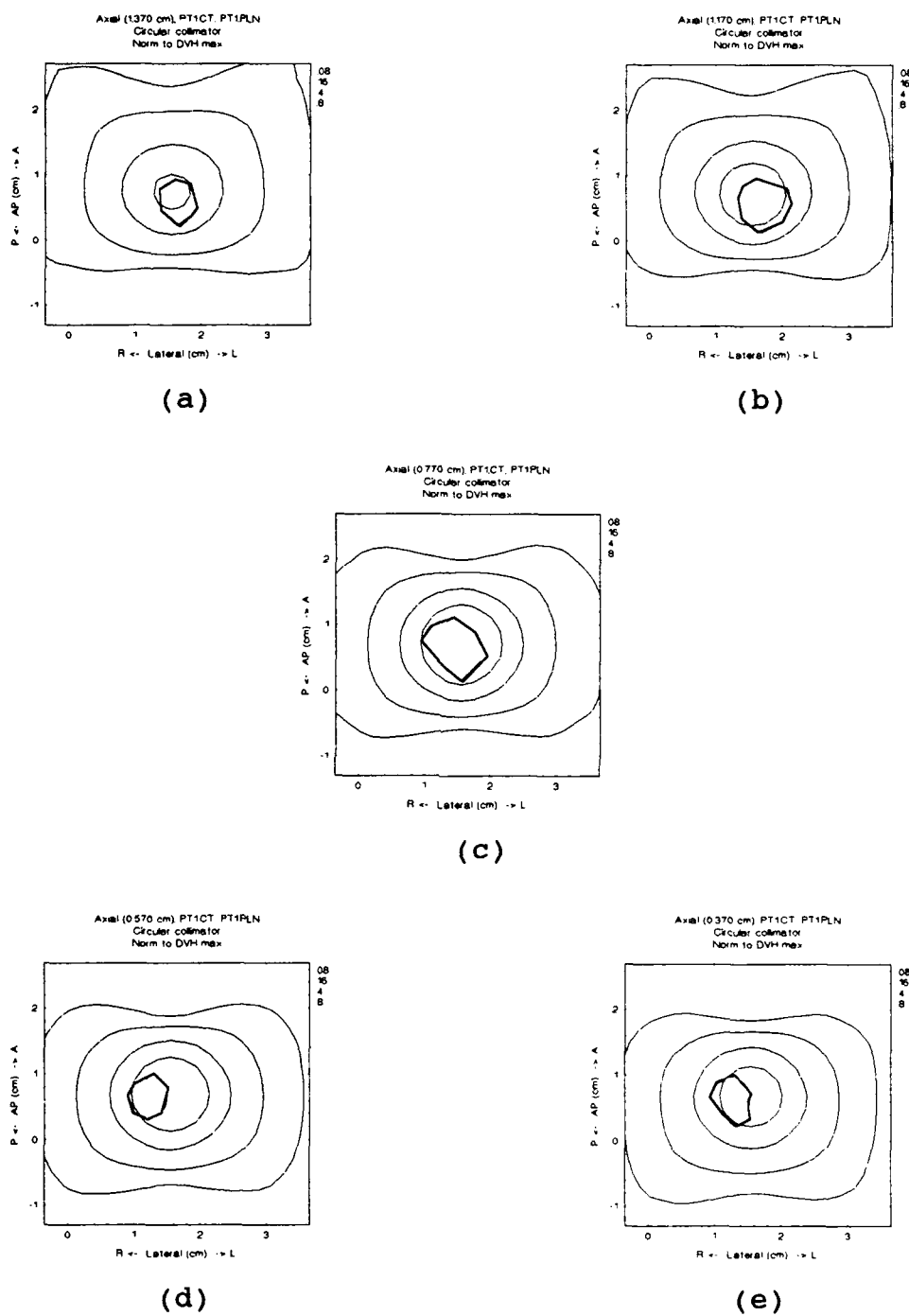
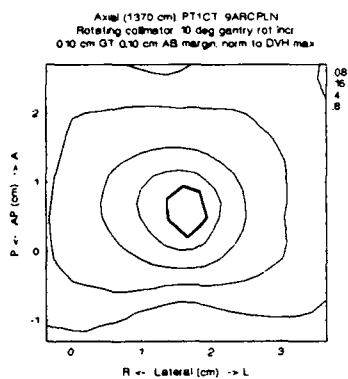
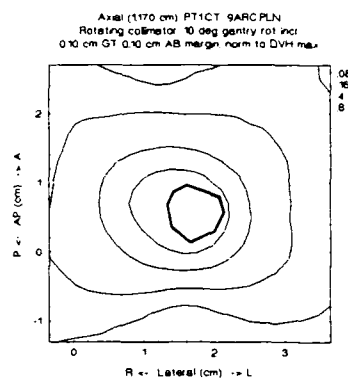


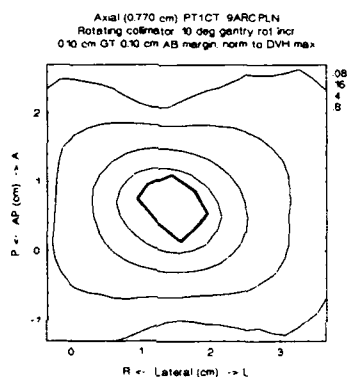
Figure 9-2: Case 1, circular collimator  
 (a) Slice 1; (b) Slice 2; (c) Slice 3  
 (d) Slice 4; (e) Slice 5



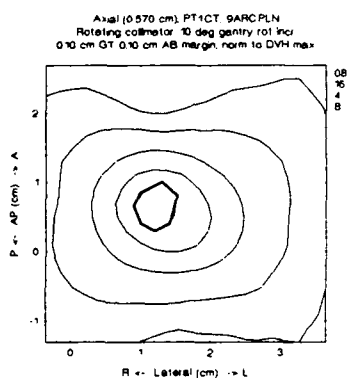
(a)



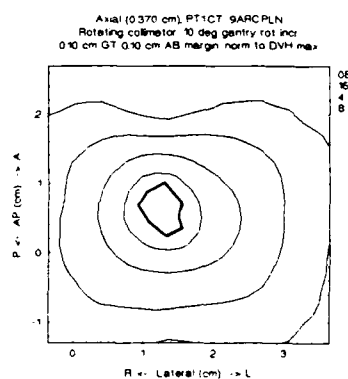
(b)



(c)

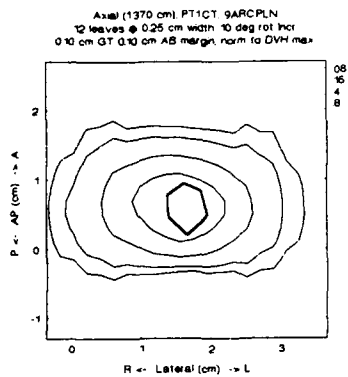


(d)

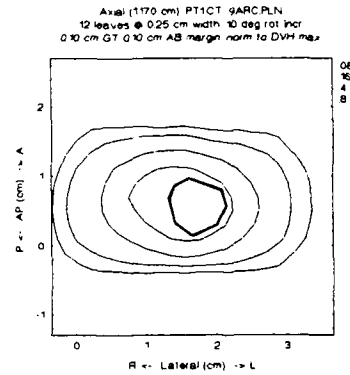


(e)

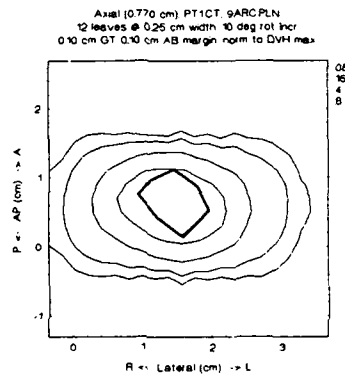
Figure 9-3: Case 1, rotating collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



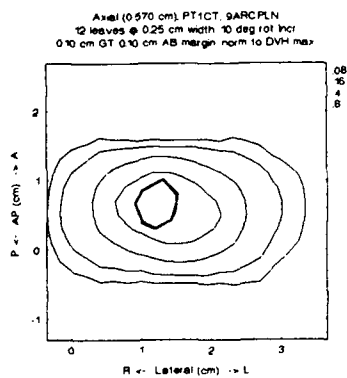
(a)



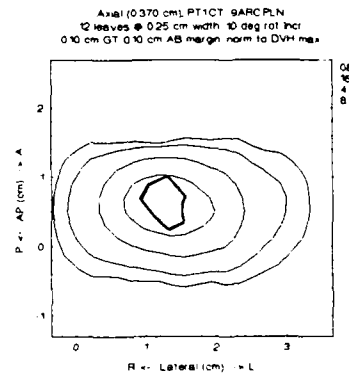
(b)



(c)

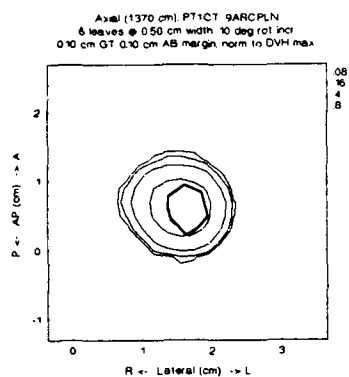


(c)

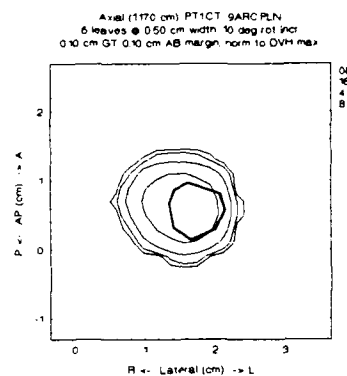


(e)

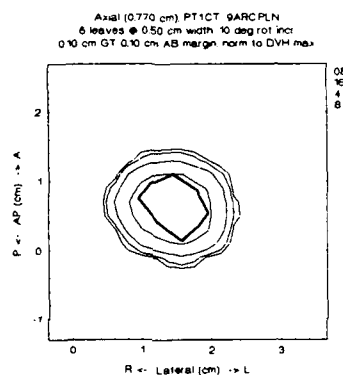
Figure 9-4: Case 1, two jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



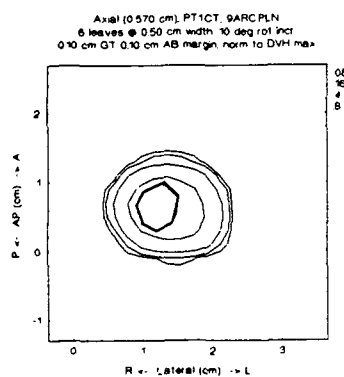
(a)



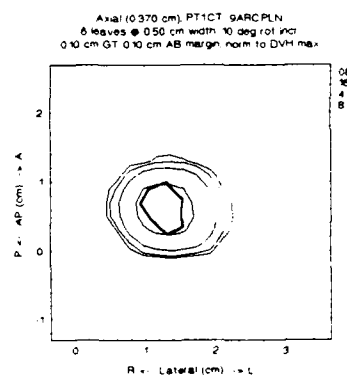
(b)



(c)



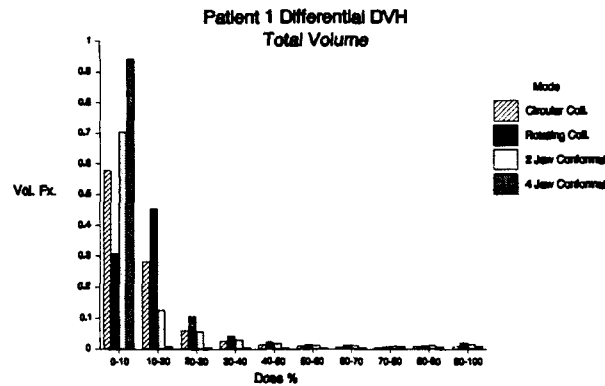
(d)



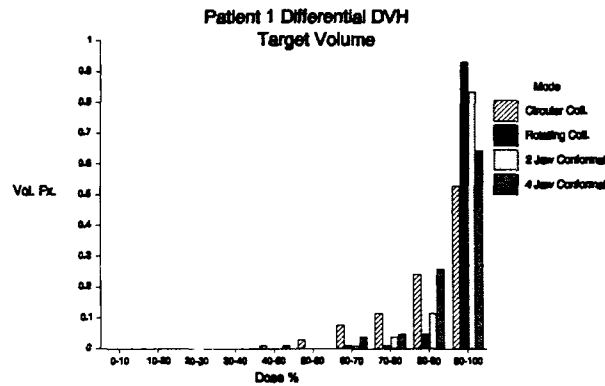
(e)

Figure 9-5: Case 1, four jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5

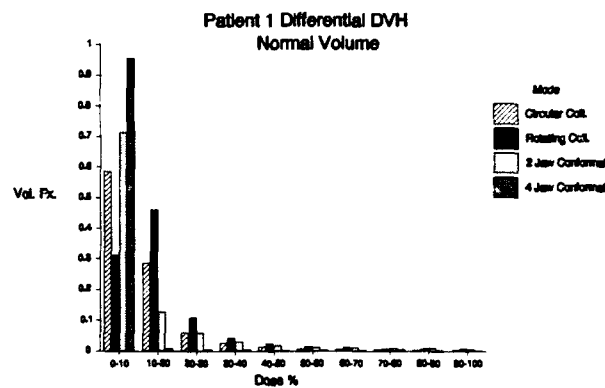




(a)

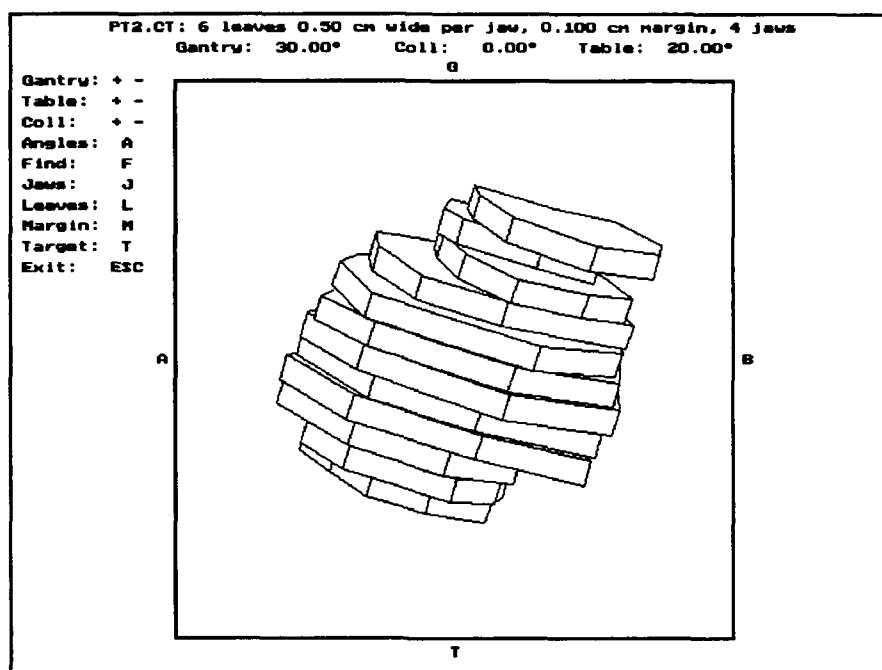


(b)

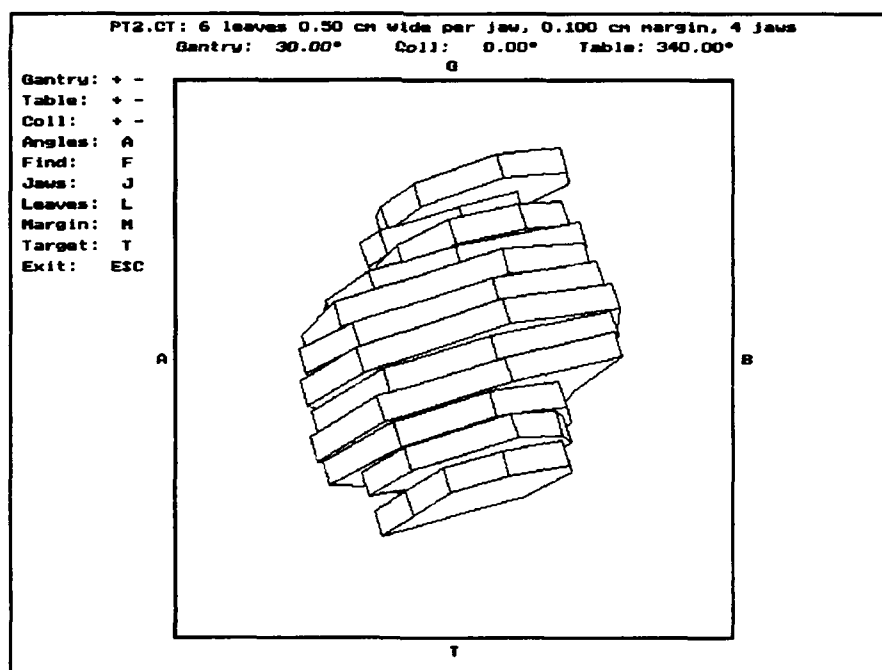


(c)

Figure 9-6: Case 1 dose volume histograms  
(a) Total volume; (b) Target volume; (c) Normal volume



(a)



(b)

Figure 9-7: Case 2 target  
 (a) View 1; (b) View 2

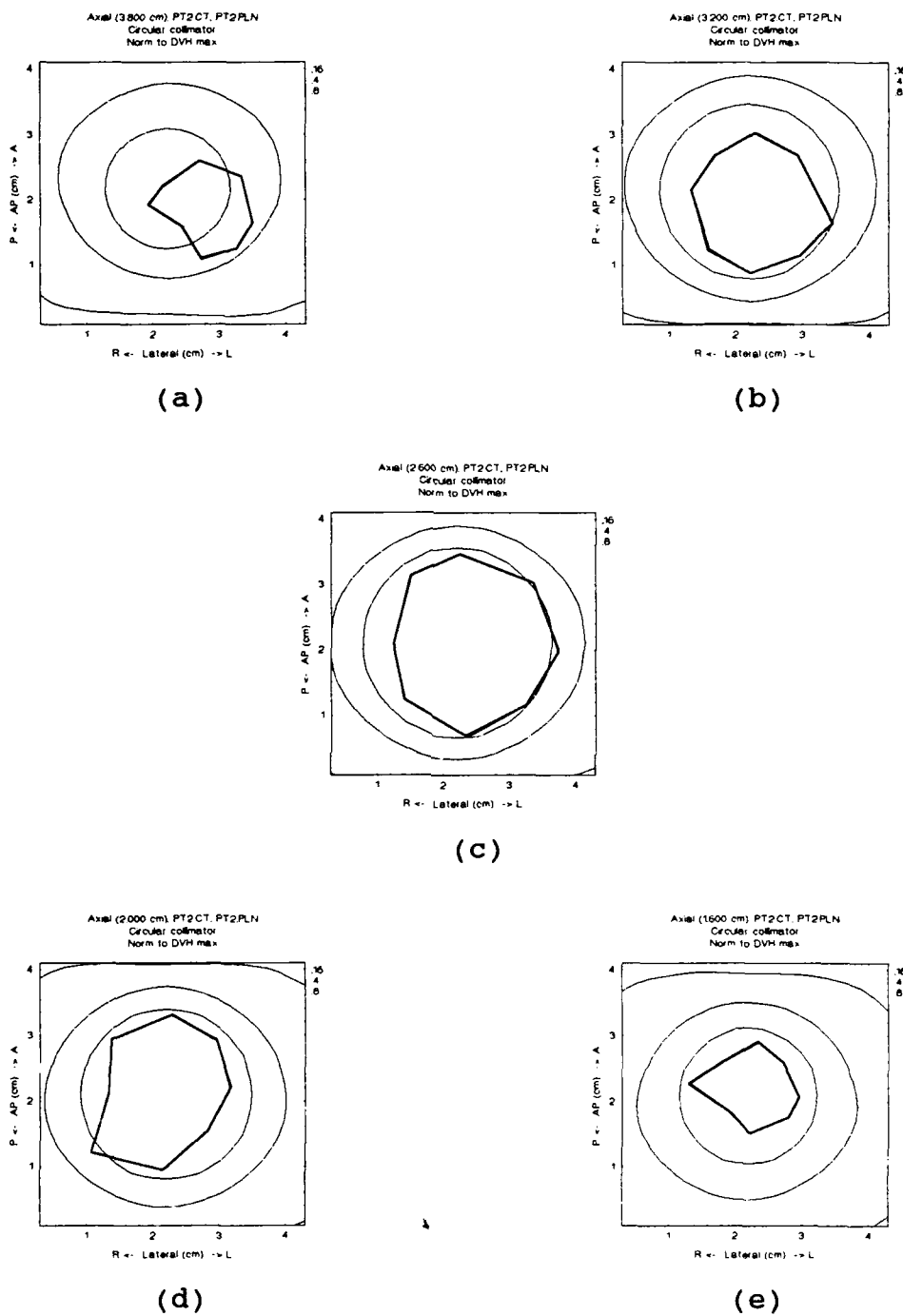
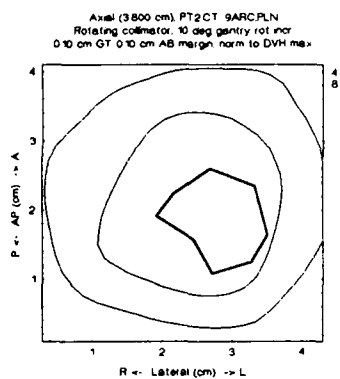
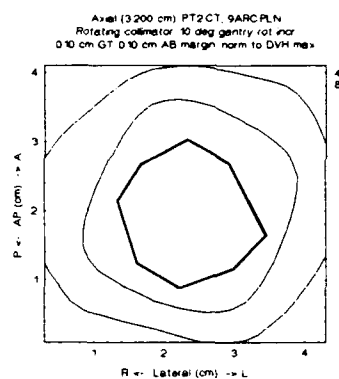


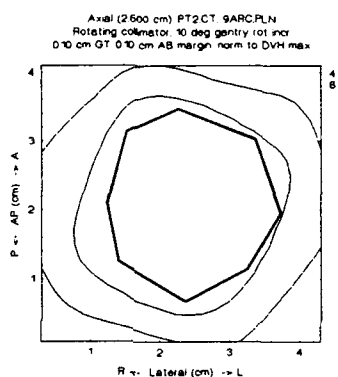
Figure 9-8: Case 2, circular collimator  
 (a) Slice 1; (b) Slice 2; (c) Slice 3  
 (d) Slice 4; (e) Slice 5



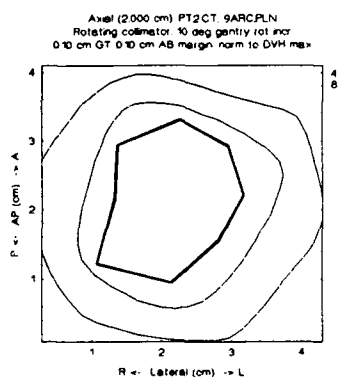
(a)



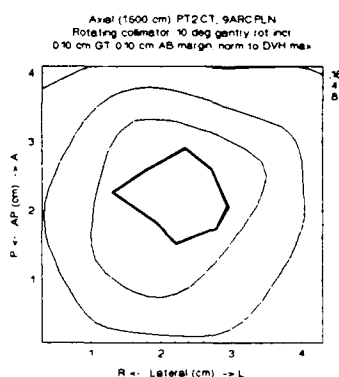
(b)



(c)

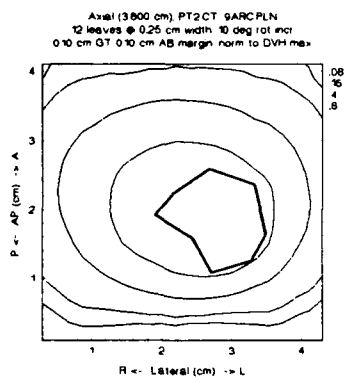


(d)

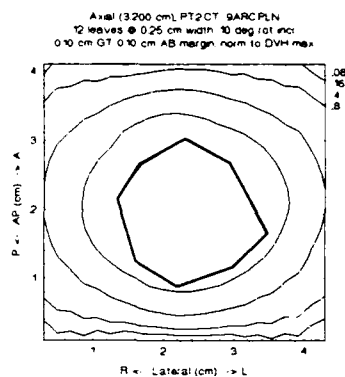


(e)

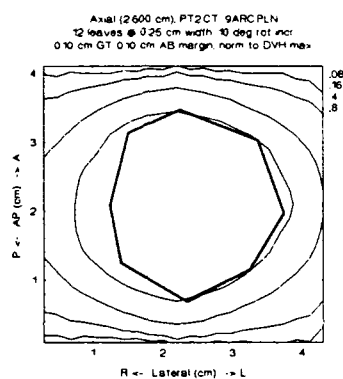
Figure 9-9: Case 2, rotating collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



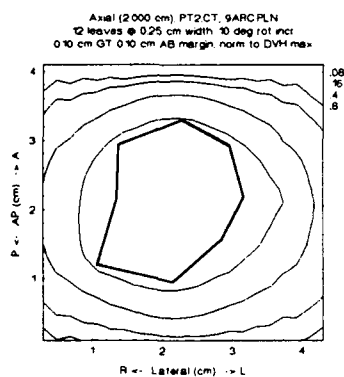
(a)



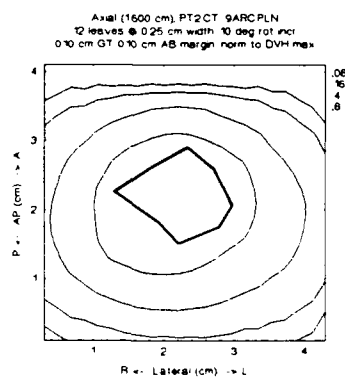
(b)



(c)

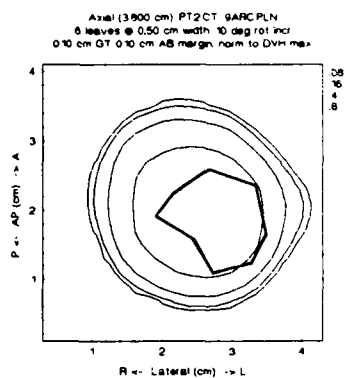


(d)

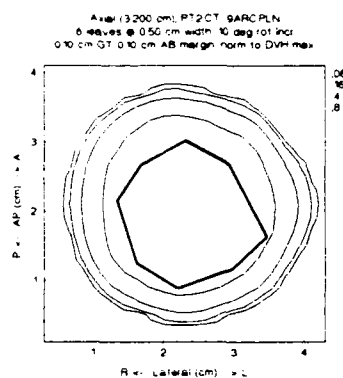


(e)

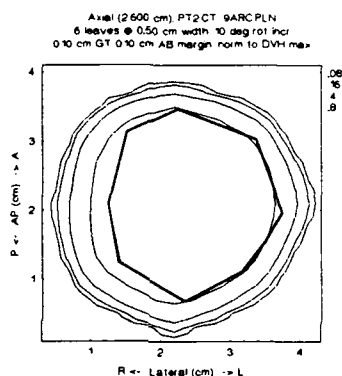
Figure 9-10: Case 2, two jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



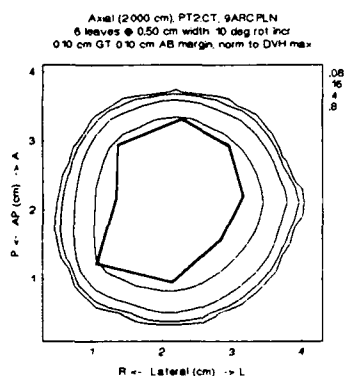
(a)



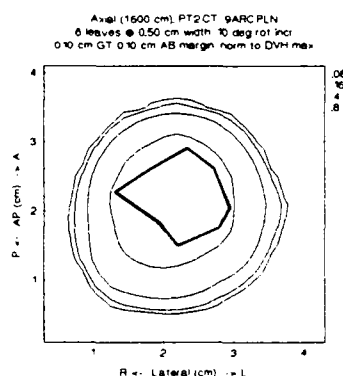
(b)



(c)

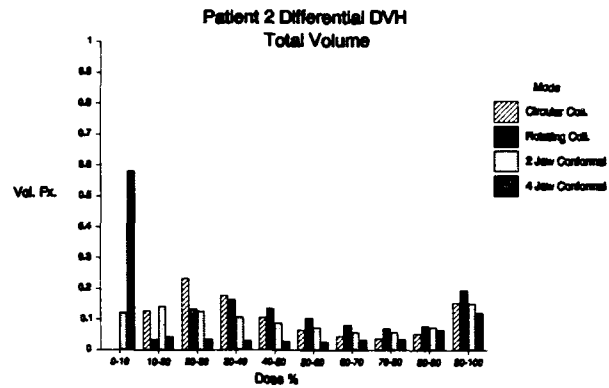


(d)

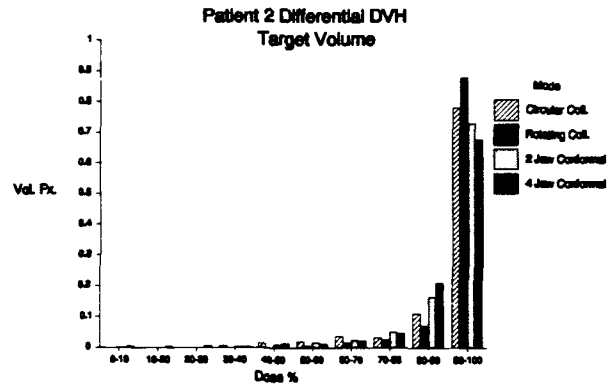


(e)

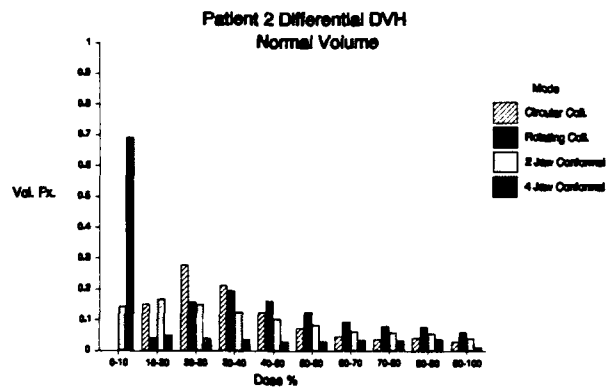
Figure 9-11: Case 2, four jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



(a)

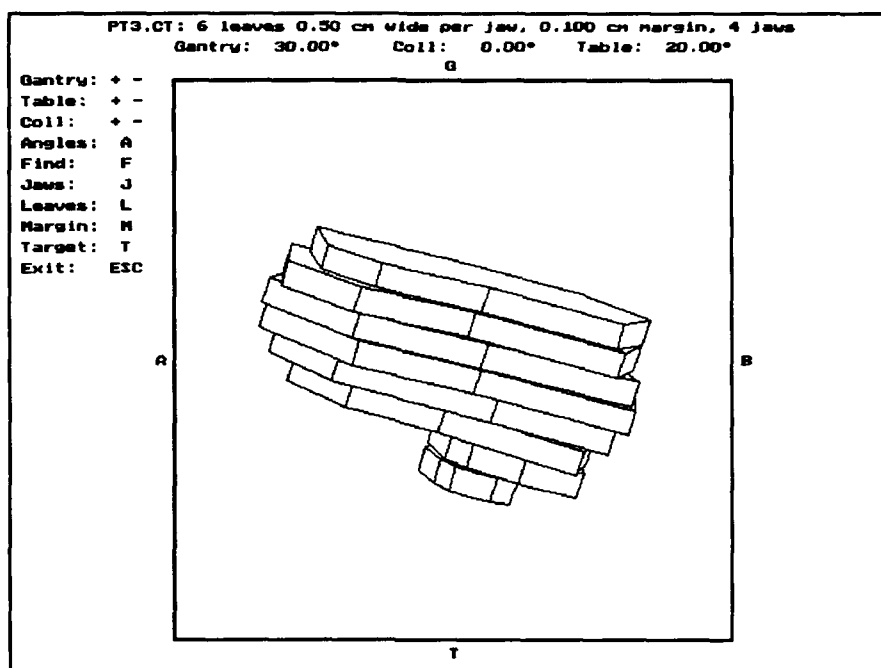


(b)

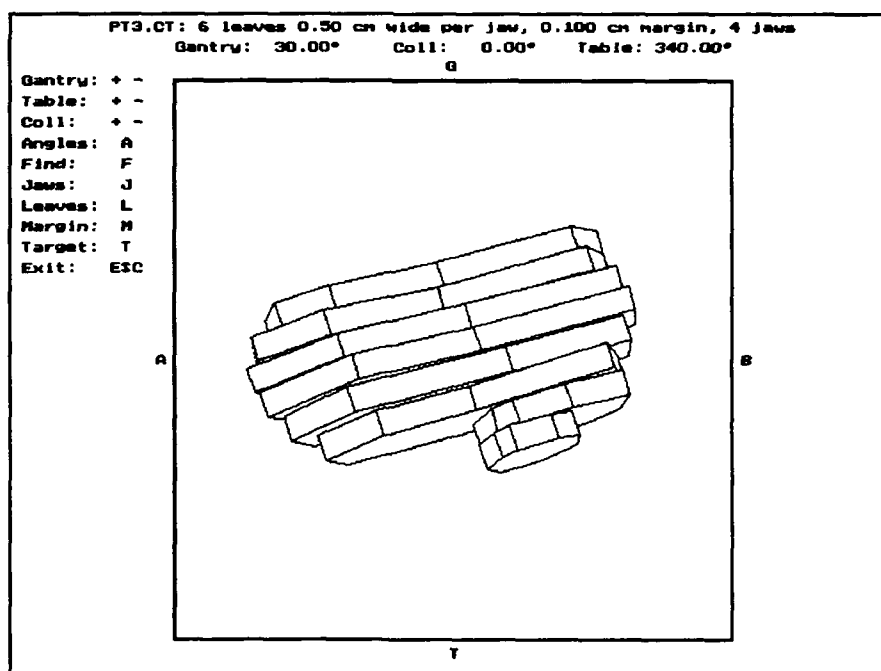


(c)

Figure 9-12: Case 2 dose volume histograms  
(a) Total volume; (b) Target volume; (c) Normal volume



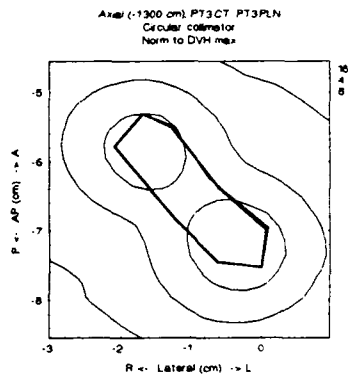
(a)



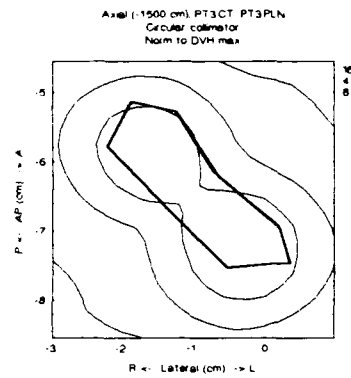
(b)

Figure 9-13: Case 3 target  
 (a) View 1; (b) View 2

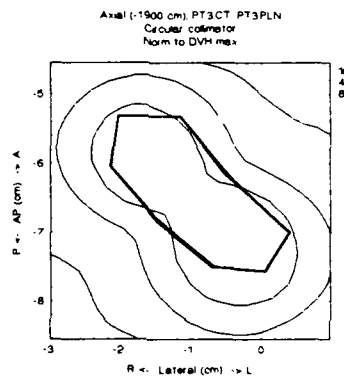




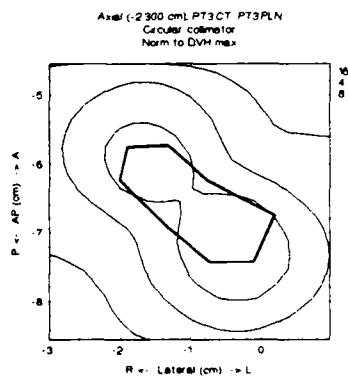
(a)



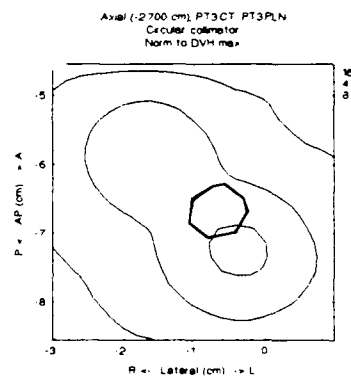
(b)



(c)

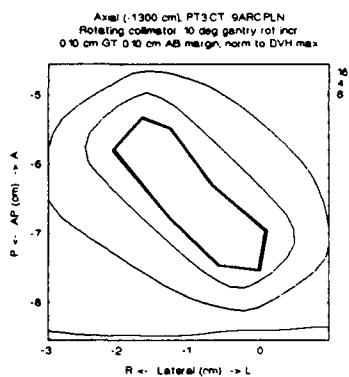


(d)

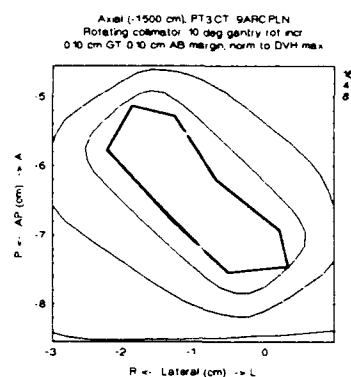


(e)

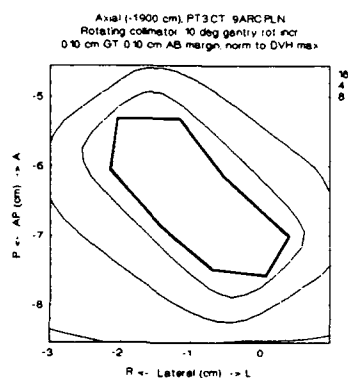
Figure 9-14: Case 3, circular collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



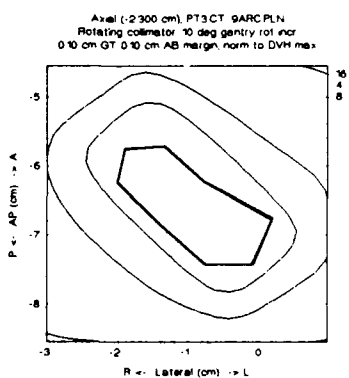
(a)



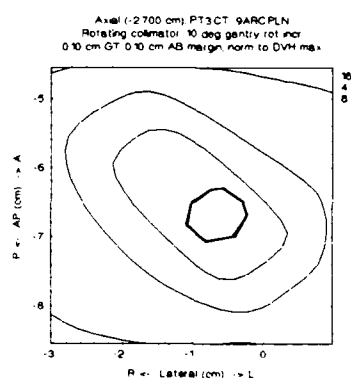
(b)



(c)

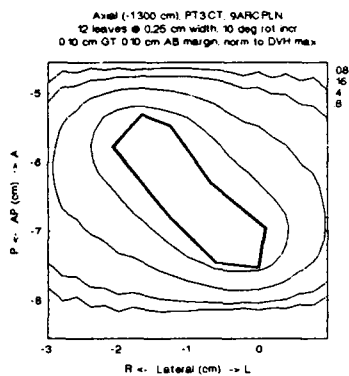


(d)

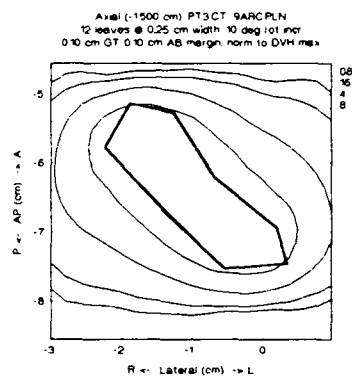


(e)

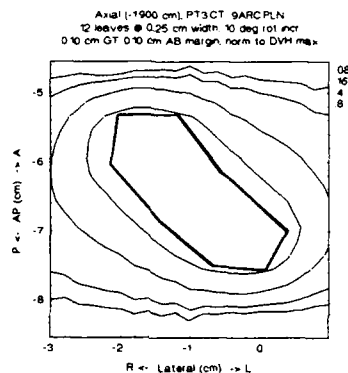
Figure 9-15: Case 3, rotating collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



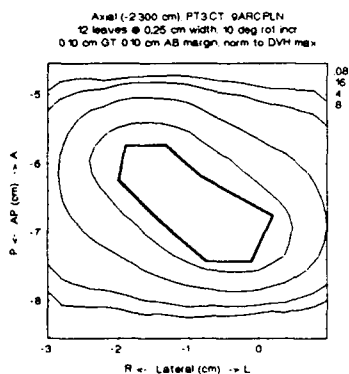
(a)



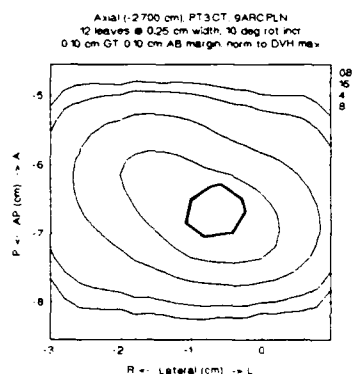
(b)



(c)

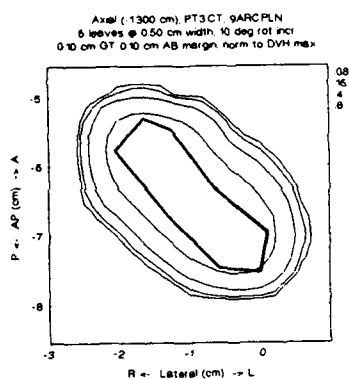


(d)

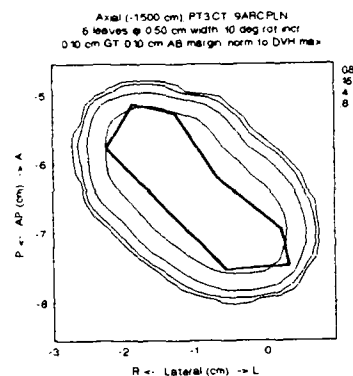


(e)

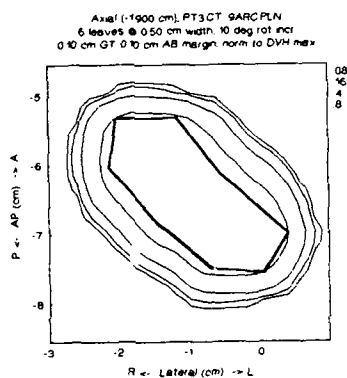
Figure 9-16: Case 3, two jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



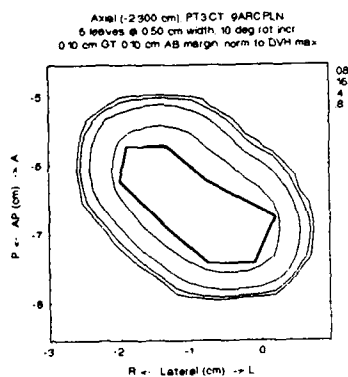
(a)



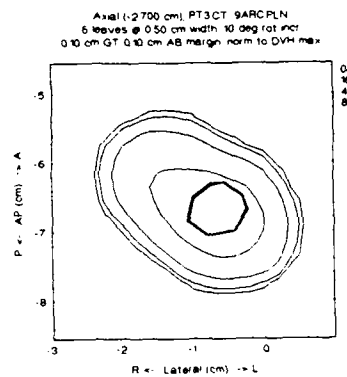
(b)



(c)

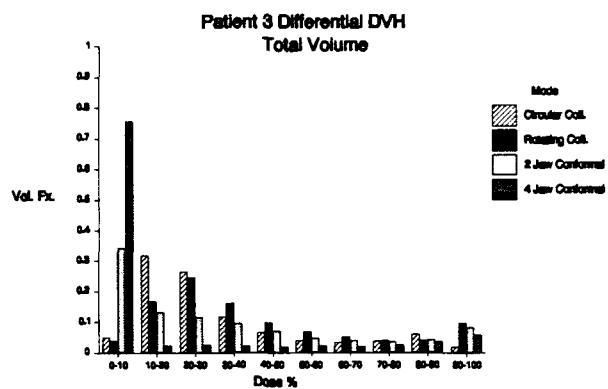


(d)

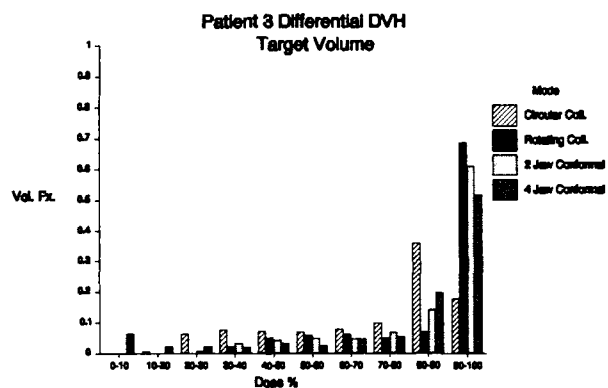


(e)

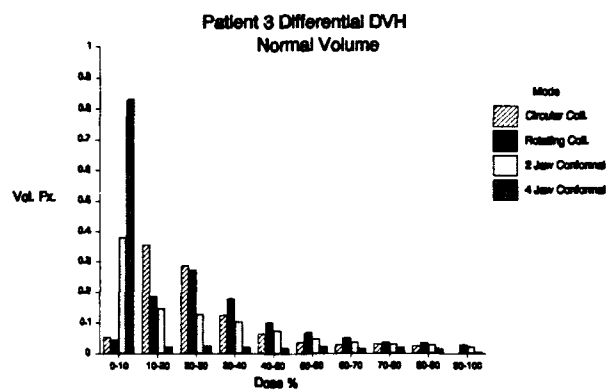
Figure 9-17: Case 3, four jaw conformal  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



(a)

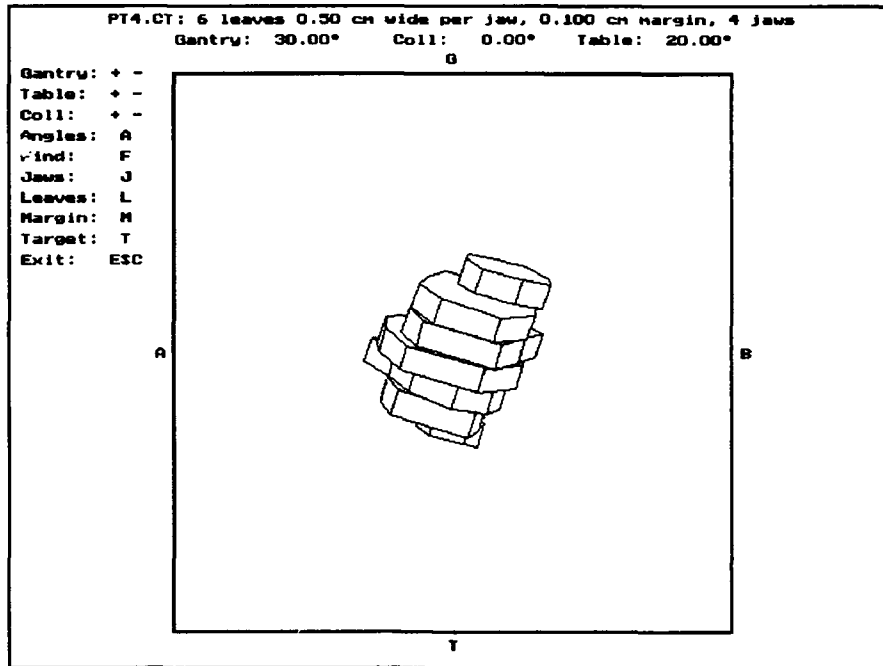


(b)

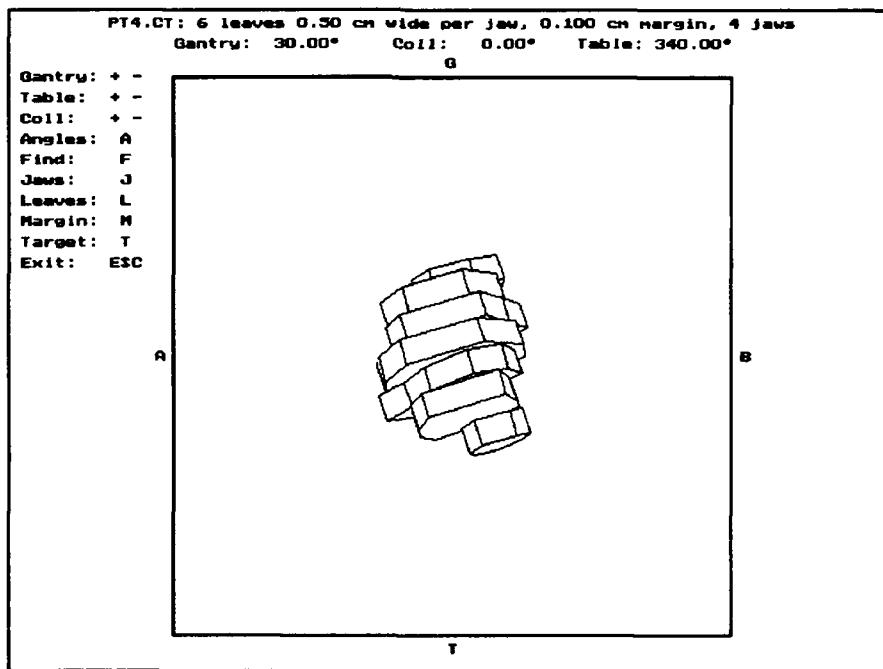


(c)

Figure 9-18: Case 3 dose volume histograms  
(a) Total volume; (b) Target volume; (c) Normal volume

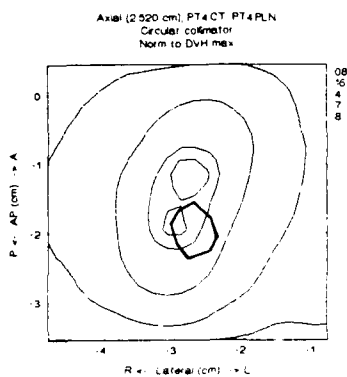


(a)

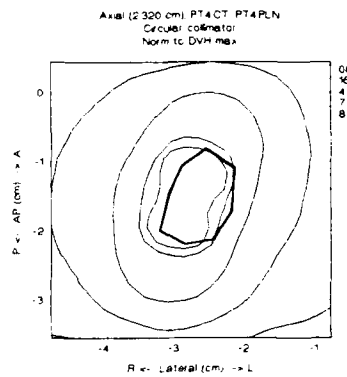


(b)

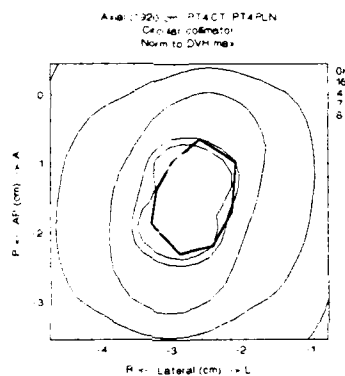
Figure 9-19: Case 4 target  
 (a) View 1; (b) View 2



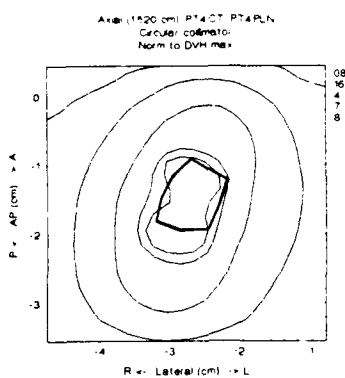
(a)



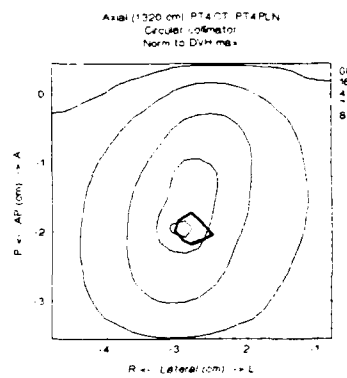
(b)



(c)

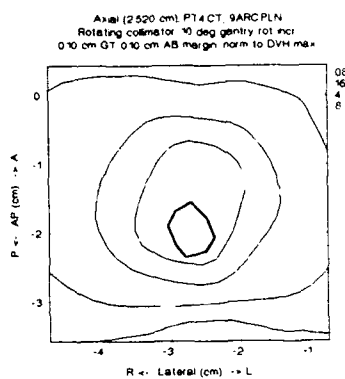


(d)

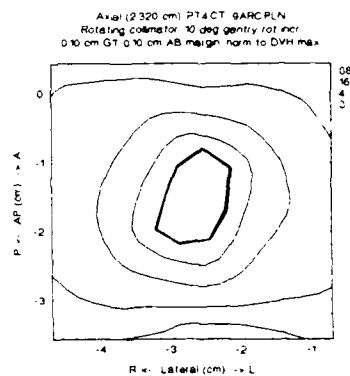


(e)

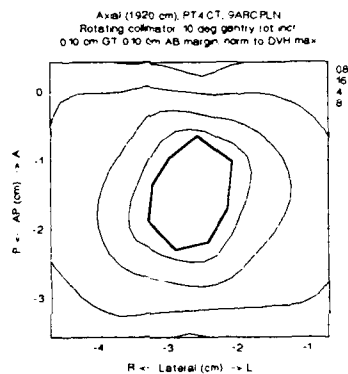
Figure 9-20: Case 4, circular collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5



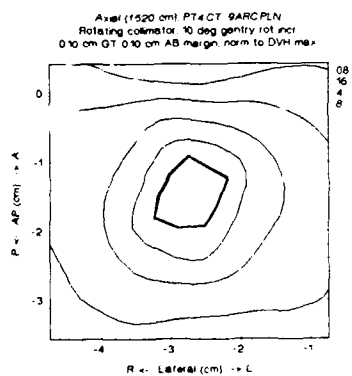
(a)



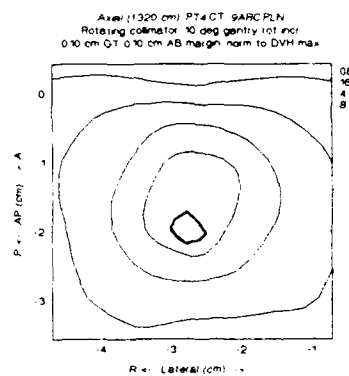
(b)



(c)



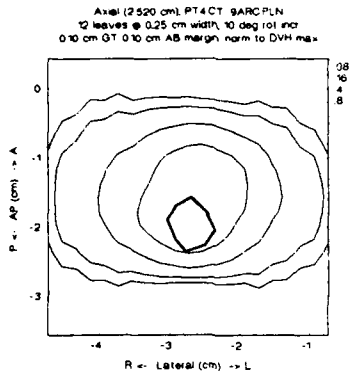
(d)



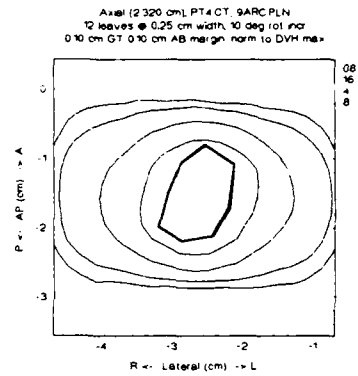
(e)

Figure 9-21: Case 4, rotating collimator  
(a) Slice 1; (b) Slice 2; (c) Slice 3  
(d) Slice 4; (e) Slice 5

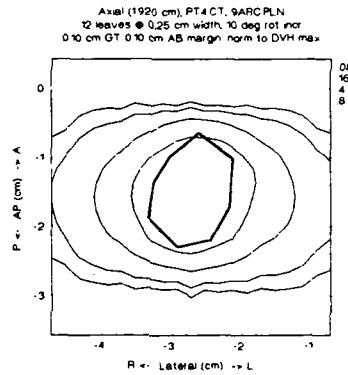




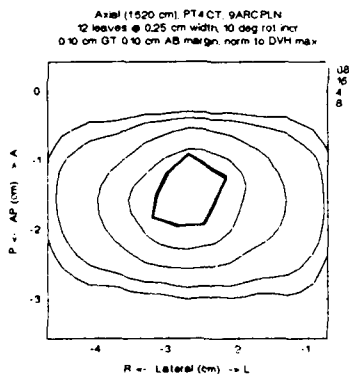
(a)



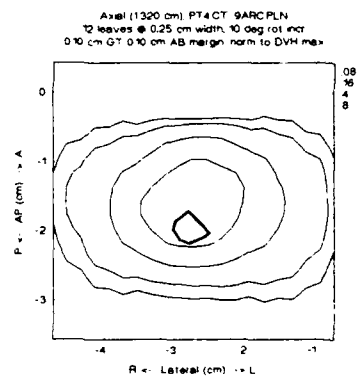
(b)



(c)



(d)



(e)

Figure 9-22: Case 4, two jaw conformal  
 (a) Slice 1; (b) Slice 2; (c) Slice 3  
 (d) Slice 4; (e) Slice 5

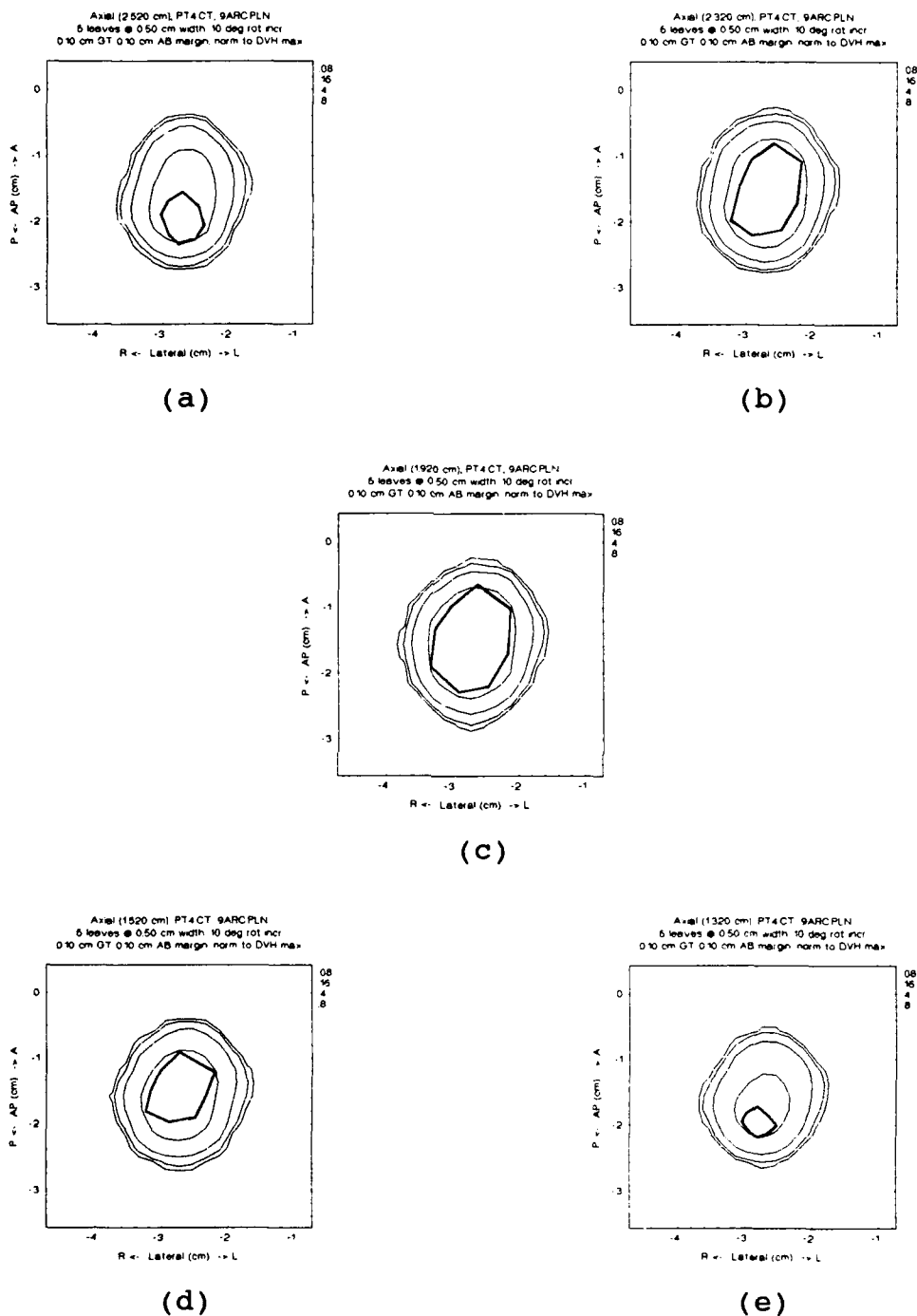
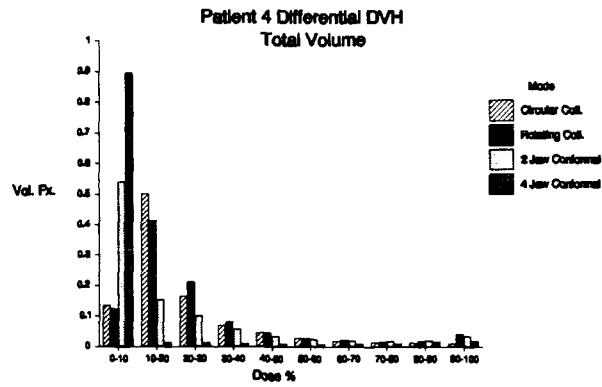
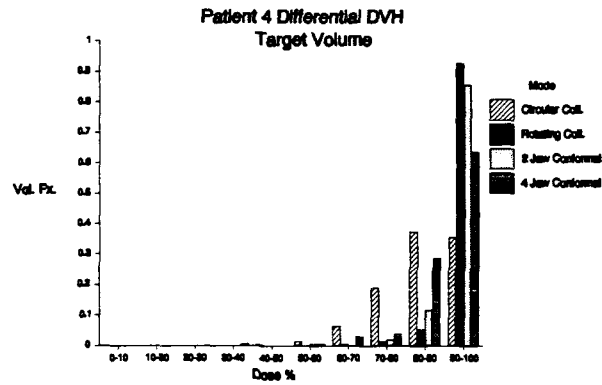


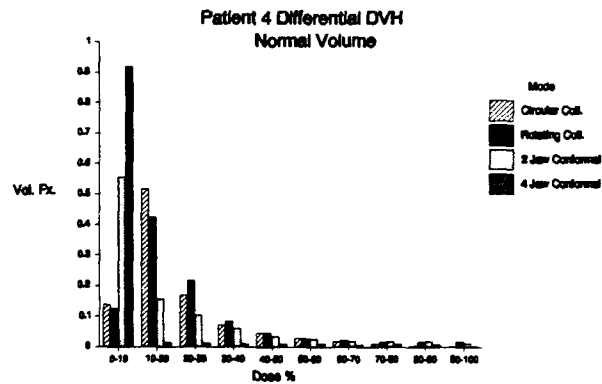
Figure 9-23: Case 4, four jaw conformal  
 (a) Slice 1; (b) Slice 2; (c) Slice 3  
 (d) Slice 4; (e) Slice 5



(a)



(b)



(c)

Figure 9-24: Case 4 dose volume histograms  
(a) Total volume; (b) Target volume; (c) Normal volume

## CHAPTER 10 CONCLUSION

This work has suggested and implemented the conceptual framework for conformal stereotactic radiosurgery using a multileaf collimator, with target localization using either one or two pairs of collimator jaws. A complete target localization and dose planning system has been developed to this end. The developed system was validated with data obtained from a working treatment machine, extensively tested for parameter optimization, and compared against two other presently existing systems for delivering radiation treatment stereotactically.

Target localization is accomplished graphically. While adherence to standard graphical function calls (such as MOVETO and LINETO which are available as standard graphic calls on a number of platforms) has been of priority, nevertheless this service is inevitably system specific (in this case, specific to IBM compatible computers with VGA graphics). As graphics functions are accomplished with integer arithmetic, this should provide the fastest possible processing of targets. Of interest in further development would be an analytical method of finding and localizing the projected area of a rotating target that would be processor

independent and comparing the speed of localization with the method developed here.

Localization is accomplished on the projected cross sectional area of a target as the gantry and table of the treatment machine rotate about the selected isocenter, thus the target is of primary importance in this system. This requires that the target contours on all axial slices in which it appears be outlined. An extension of this would be to also outline any sensitive structures and modify the localization program to specifically shield these structures, with appropriate priority weighting. This would entail a several parameter optimization approach and would focus on the lower isodose region overlap, as this work has demonstrated that it is possible to restrict the high dose regions to the immediate boundary of the target.

Two methods of three dimensional irregular field dosimetry have been investigated. One, the FFT method, while interesting from a mathematical point of view, has been shown to be inappropriate due to exceptionally long processing time when used as a three dimensional tool. The method developed here, the modified negative field method, is accurate (demonstrated average error of less than 3% or 0.5 mm for dose levels above 40% of maximum) and reasonably fast, but employs a painstaking exact dose calculation on every collimator leaf in every field. Various approximation methods could be investigated to further speed processing time.

As this work was conceptual only, of obvious interest would be the construction of a physical multileaf collimator device that could be attached to the present dose delivery system at the University of Florida. Such a device must be compact to fit in the limited space available under the gantry that presently holds the fixed circular collimator. Since a four jaw (two pair/two level) multileaf collimator must be twice the length of a two jaw (one pair/one level) device, and as an analysis of the volume dosimetry comparing these shows only a small performance degradation of the two jaw with respect to the four jaw, and that degradation in the low dose region, it is recommended that the two jaw approach be used to both save space and reduce complexity, not only of the device itself, but of the dosimetry and the interface to the treatment machine. Careful measurements of the resulting machine output must be made on any realized device and these data must be incorporated into the dosimetry software.

In conclusion, this work has presented an effective system to conformally treat small targets with stereotactic radiosurgery. This system should be of advantage, with respect to the current treatment modes in use, in planning and treating targets of irregular shape that are now planned and treated with difficulty.

## APPENDIX A TREATMENT PLANNING

The complete treatment planning process developed in this study is shown in the flow chart, figure A-1. The process flows from program LFTP (treatment plan input, listed in this appendix) to program LFLOC (target location, appendix B). Program LFDOS (plane dosimetry, appendix C) and/or program LFDVH (dose volume histogram, appendix D) may then be run. Evaluation of the dose volume histogram may be made with program ILFN (the integrated logistic function, appendix E). Prior to planning, it is assumed that the target has been outlined on each axial CT slice, and has been saved in a data file of the form:

```
1) 0.15
2) 8
3) 7
4) 0.5 0.5 0.5
5) 0.25 0.25 0.5
.
.
.
```

Note that the line numbers are not included in the data file and are only used as reference in the following description. Line 1 is the CT slice thickness in centimeters. Line 2 is the number of data points in each slice. Line 3 is the number of slices in which the target occurs. Lines 4 through the end of the file are the AP, lateral, and axial

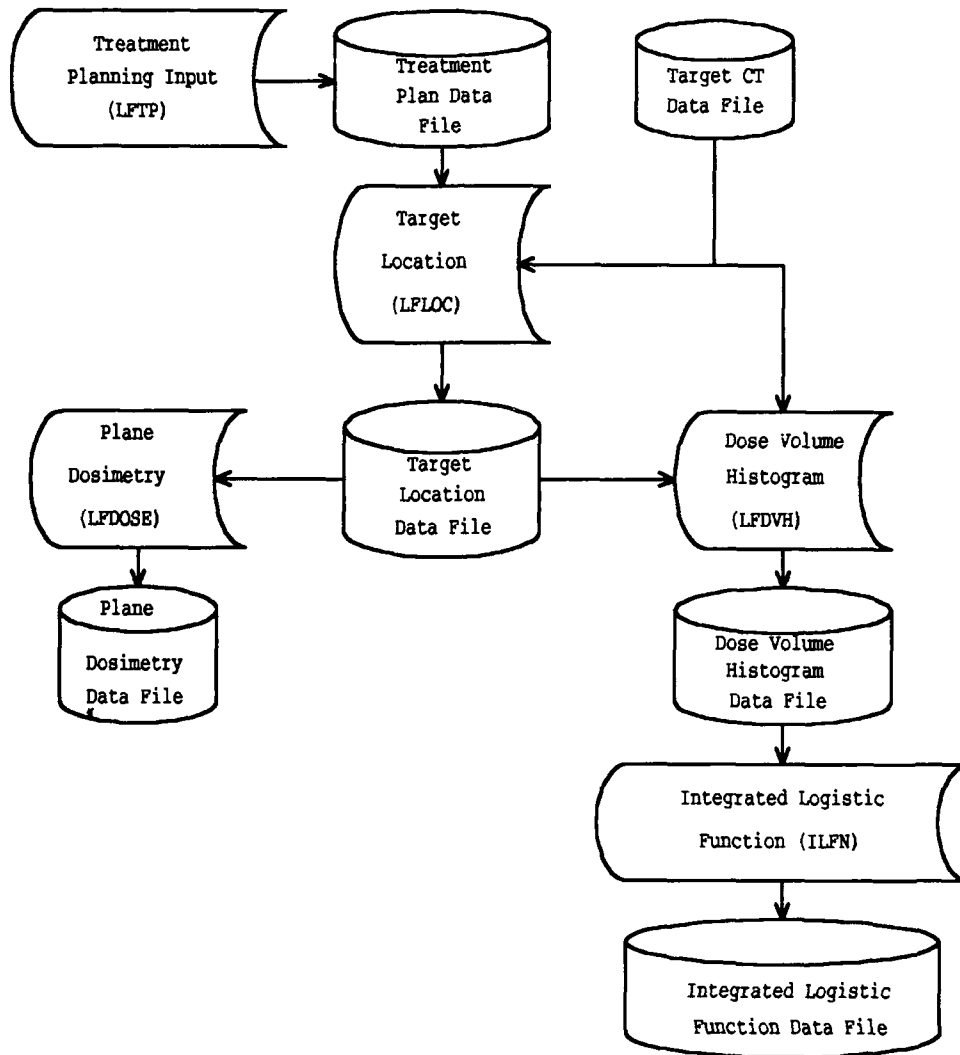


Figure A-1: Treatment planning process



BRW coordinates of each point, respectively, outlining the target on each axial slice.

The program LFTP.C interactively creates a data file to be used as input to LFLOC.C, the localization program, as the first step in the conformal dosimetry process. All input is explicitly requested. The input for the treatment plan offers the choice of either predefined treatment plan files or individual inputs for the parameters in a user defined plan.

The output file is a tab delimited, ASCII data file and has the form:

1)	sphere.ct			
2)	5ARC.PLN			
3)	5			
4)	0.000	0.000	0.000	
5)	6	0.500	0.05000	
6)	0.000	0.000	0.000	
7)	-2.000	2.000		
8)	-2.000	2.000		
9)	-2.000	2.000		
10)	0.200			
11)	10.000			
12)	10.0	30.0	130.0	1.000
13)	50.0	30.0	130.0	1.000
14)	350.0	230.0	330.0	1.000
15)	310.0	230.0	330.0	1.000
16)	270.0	230.0	330.0	1.000
17)	0.1000	0.1000		

Note that the line numbers are not included in the file and are only used as reference for the following description. Line 1 is a character string (case insensitive) with the name of the target CT file. Line 2 is a character string (also case insensitive) with the name of the treatment plan file (prepared files of standard treatment plans,

used for convenience). Line 3 is an integer, the number of arcs. Line 4 are three float values of the BRW rotation point (AP, lateral, and axial or vertical in centimeters). Line 5 is an integer and two float values denoting the number of leaves per collimator side, the leaf width in centimeters, and the leaf transmission factor. Line 6 has three float values of the AP, lateral, and axial center point of the dose computation matrix in centimeters. Lines 7, 8, and 9 each have two float values representing the spacial limits, in centimeters, of the dose calculation matrix in the AP, lateral, and axial directions, respectively. If any one set of these pairs are equal, a cross plot at that coordinate is produced. Line 10 is a float value of the dose calculation matrix spacing in centimeters. Line 11 is a float value for the gantry rotation increment in degrees. Lines 12 through 16 (in this case for 5 arcs) contain the treatment plan named in line 2. All values are float and denote, in order, the table angle in degrees, the gantry start and stop angles in degrees, and the arc weight. Line 17 contains two float values for the GT and AB side target margins in centimeters.

```

// Program LFTP.C
//   To interactively create a treatment plan data file for
// use in program LFLOC.C

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>

float **matrix(int rl, int rh, int cl, int ch);
void free_matrix(float **m, int rl, int rh, int cl);

void main(void)
{
    int i, arcs, lvs;

    float aprp, larp, axrp, lwid, tran, ap0, la0, ax0, ap1,
          ap2, la1, la2, ax1, ax2, del, rinc, gtmargin,
          abmargin, **pm;

    char *infile, *outfile, *ctfile, plan, *tplan;

    FILE *infp, *outfp;

    infile = (char *)malloc(12 * sizeof(char));
    outfile = (char *)malloc(12 * sizeof(char));
    ctfile = (char *)malloc(12 * sizeof(char));
    tplan = (char *)malloc(12 * sizeof(char));

    printf("Enter CT target file name: ");
    scanf("%s", ctfile);

    printf("Enter the number of arcs: ");
    scanf("%i", &arcs);

    printf("Enter the target rotation center (cm):\n");
    printf("      AP: ");
    scanf("%f", &aprp);
    printf("      Lateral: ");
    scanf("%f", &larp);
    printf("      Axial: ");
    scanf("%f", &axrp);

    printf("Enter the number of collimator leaves: ");
    scanf("%i", &lvs);

    printf("Enter the leaf width (cm): ");
    scanf("%f", &lwid);

    printf("Enter the leaf transmission factor: ");
    scanf("%f", &tran);

```

```

printf("Enter the dose matrix zero point (cm):\n");
printf("      AP: ");
scanf("%f", &ap0);
printf("      Lateral: ");
scanf("%f", &la0);
printf("      Axial: ");
scanf("%f", &ax0);

printf("Enter the dose matrix limits (cm):\n");
printf("      AP1 AP2: ");
scanf("%f %f", &ap1, &ap2);
printf("      Lateral1 Lateral2: ");
scanf("%f %f", &la1, &la2);
printf("      Axial1 Axial2: ");
scanf("%f %f", &ax1, &ax2);

printf("Enter the matrix spacing (cm): ");
scanf("%f", &del);

printf("Enter the gantry rotation increment (deg): ");
scanf("%f", &rinc);

printf("Enter S for standard plan, C for custom plan: ");
plan = getche();
plan = toupper(plan);
pm = matrix(1, arcs, 1, 4);
if(plan == 'S')
{
    printf("\n Enter plan file: ");
    scanf("%s", infile);
    if((infp = fopen(infile, "r")) == NULL)
    {
        printf("Input file %s unavailable\n", infile);
        exit(0);
    }
    for(i = 1; i <= arcs; i++)
        fscanf(infp, "%f %f %f %f", &pm[i][1], &pm[i][2],
            &pm[i][3], &pm[i][4]);

    fclose(infp);

    strcpy(tplan, strupr(infile));

    free(infile);
}
else
{
    for(i = 1; i <= arcs; i++)
    {
        printf("\n Arc %i:\n", i);
        printf("      Table angle (deg): ");
        scanf("%f", &pm[i][1]);
    }
}

```

```

    printf("    Gantry start (deg): ");
    scanf("%f", &pm[i][2]);
    printf("    Gantry stop (deg): ");
    scanf("%f", &pm[i][3]);
    printf("    Arc weight: ");
    scanf("%f", &pm[i][4]);
}
strcpy(tplan, "USRDEF.PLN");
}

printf("Enter the GT collimator margin (cm): ");
scanf("%f", &gtmargin);

printf("Enter the AB collimator margin (cm): ");
scanf("%f", &abmargin);

printf("\nEnter output file: ");
scanf("%s", outfile);
if((outfp = fopen(outfile, "w")) == NULL)
{
    printf("Output file %s cannot be opened\n", outfile);
    exit(0);
}

fprintf(outfp, "%s\n", ctfile);
fprintf(outfp, "%s\n", tplan);
fprintf(outfp, "%i\n", arcs);
fprintf(outfp, "%.3f\t%.3f\t%.3f\n", aprp, larp, axrp);
fprintf(outfp, "%i\t%.3f\t%.3f\n", lvs, lwid, tran);
fprintf(outfp, "%.3f\t%.3f\t%.3f\n", ap0, la0, ax0);
fprintf(outfp, "%.3f\t%.3f\n", ap1, ap2);
fprintf(outfp, "%.3f\t%.3f\n", la1, la2);
fprintf(outfp, "%.3f\t%.3f\n", ax1, ax2);
fprintf(outfp, "%.3f\n", del);
fprintf(outfp, "%.3f\n", rinc);
for(i = 1; i <= arcs; i++)
    fprintf(outfp, "%.1f\t%.1f\t%.1f\t%.3f\n", pm[i][1],
                                                    pm[i][2],
                                                    pm[i][3],
                                                    pm[i][4]);

fprintf(outfp, "%.4f\t%.4f\n", gtmargin, abmargin);

fclose(outfp);

free(outfile);
free(ctfile);
free(tplan);
free_matrix(pm, 1, arcs, 1);
}

// allocates a float matrix with range [rl..rh][cl..ch]
// from "Numerical Recipes in C" [Pre88]

```

```

float **matrix(int rl, int rh, int cl, int ch)
{
    int i;
    float **m;

    // allocate pointers to rows
    m = (float **)malloc((unsigned)(rh - rl + 1) *
                          sizeof(float *));
    if(m == NULL)
    {
        printf("Allocation failure 1 in matrix()");
        exit(0);
    }
    m -= rl;

    // allocate rows and set pointers to them
    for(i = rl; i <= rh; i++)
    {
        m[i] = (float *)malloc((unsigned)(ch - cl + 1) *
                                sizeof(float));
        if(m[i] == NULL)
        {
            printf("Allocation failure 2 in matrix()");
            exit(0);
        }
        m[i] -= cl;
    }

    // return pointer to array of pointers to rows
    return m;
}

// frees a matrix allocated with matrix()
// from "Numerical Recipes in C" [Pre88]
void free_matrix(float **m, int rl, int rh, int cl)
{
    int i;

    for(i = rh; i >= rl; i--)
        free((char *)(m[i] + cl));
    free((char *)(m + rl));
}

```

## APPENDIX B TARGET LOCALIZATION PROGRAM

The target localization program, LFLOC.C, finds the leaf positions to conform the multileaf collimator to the projected outline of the target using the parameters supplied by the data file generated by LFTP.C, as described in appendix A. This process is illustrated in the program flow chart presented in figure B-1. The command line for the target localization program takes three arguments; the input file name, the output file name, and the number of jaws with which to localize (2 or 4).

The input file for the target localization program is generated by answering the questions posed by the treatment plan program LFTP.C. The LFTP program then writes an appropriate data file. This file contains all that the LFLOC program needs to function, including the CT target file and the treatment plan prescription, as well as calculation parameters. The output file contains all that either dosimetry program (plane or dose volume histogram) needs to calculate the required dosimetry.

The output file has the form:

- 1) sphere.ct
- 2) 5ARC.PLN
- 3) 0.10 0.10
- 4) 5

```

5)      0.000 0.000 0.000
6)      6 0.500 0.05000
7)      10.000
8)      10.000 30.000 130.000 1.000
9)      50.000 30.000 130.000 1.000
10)     350.000 230.000 330.000 1.000
11)     310.000 230.000 330.000 1.000
12)     270.000 230.000 330.000 1.000
13)     0.000 0.000 0.000
14)     -2.000 2.000
15)     -2.000 2.000
16)     -2.000 2.000
17)     0.2000
18)     0.4567
19)     -1.50  0.65  0.79  0.77  0.68 -1.50
        -1.50  0.62  0.79  0.78  0.62 -1.50
        -1.50 -0.64 -0.79 -0.77 -0.67 -1.50
        -1.50 -0.63 -0.80 -0.80 -0.63 -1.50
        .
        .
        .

```

Note that the line numbers do not appear in the output file, but are used here for the file description. Line 1 is a case insensitive character string holding the name of the CT file used to localize the target. Line 2 is a case insensitive character string holding the name of the treatment plan. Line 3 has two float values indicating the GT and AB target localization margins in centimeters. Line 4 is an integer showing the number of arcs. Line 5 is the BRW rotation point for the AP, lateral, and axial or vertical axes, respectively, in centimeters. Line 6 has an integer representing the number of leaves per collimator side, and two float values for the leaf width in centimeters and the leaf transmission factor. Line 7 is a float value for the gantry rotation increment in degrees. Lines 8 through 12 (in this case) are float values for the treatment plan with



each line having entries for the table angle in degrees, the gantry start and stop angles in degrees, and the arc weight. Line 13 is the matrix cutting point in centimeters for the AP, lateral, and vertical axes, respectively. Lines 14 through 16 denote the matrix calculation limits in centimeters for the AP, lateral and vertical axes. Line 17 is the dosimetry matrix spacing in centimeters. Line 18 is the equivalent area/perimeter of the conformed field. Line 19 is a series of float values, each of which denotes the position of the leading edge of a leaf into the open field of the collimator. The values represent each leaf from the gantry side in the GA position clockwise. Lines 18 and 19 form a pair which is repeated for every gantry/table position in the treatment plan. In this case, five arcs of  $100^\circ$  each with a  $10^\circ$  gantry increment result in 50 leaf position specification lines, with only the first shown.

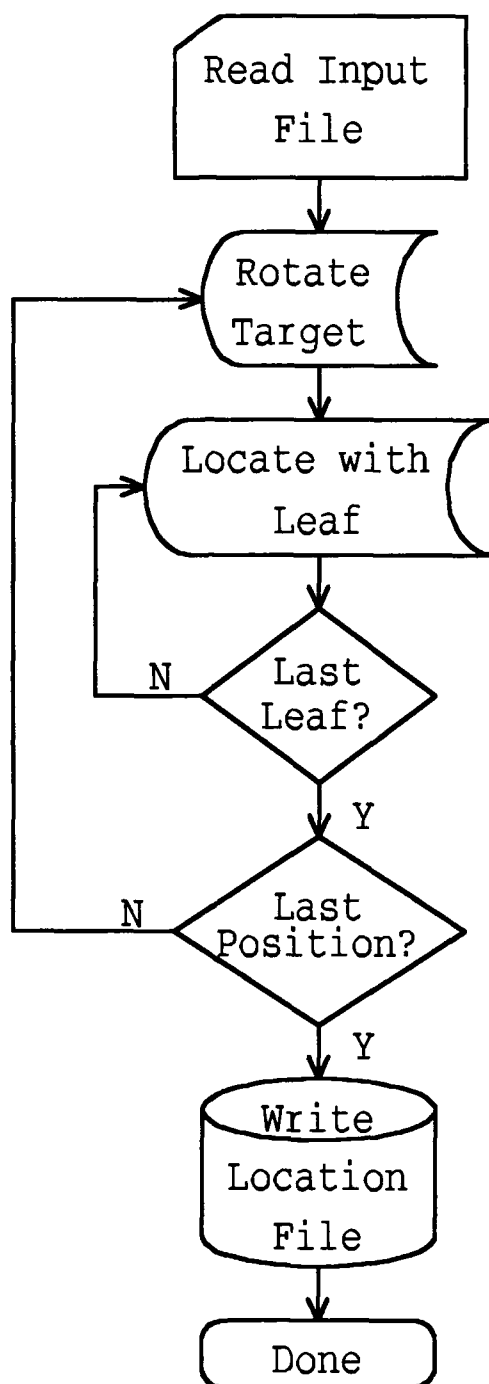


Figure B-1: LFLOC.C flow chart

```

// Program LFLOC.C
//   To locate the projected area of a target defined in a
//   CT file
//   Uses the output of program LFTP.C as the data input
//   Input is either interactive or on the command line
//   Command line LFLOC <input file> <output file>
//   <collimator jaws>
//   Output file is used by both LFDOSE.C and LFDVH.C
//   Collimator jaws either 2 or 4
//   This program will work on a VGA system only

#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <string.h>
#include <dos.h>
#include <ctype.h>
#include <graphics.h>

void polydraw(int *x, int *y, int pts);
void tile(int *x1, int *y1, int *x0, int *y0, int pts);
int dispX(float xpt, int wide, float window);
float realX(int posit, int wide, float window);
int dispY(float ypt, int high, float window);
float realY(int posit, int high, float window);
int round(float in);
int prec(double index, int pl);
char *ftoa(float fin, int place);
int limit(int in, int lo, int hi);

// set stack size to 32K
extern unsigned _stklen = 32767U;

void main(int argc, char *argv[])
{
    char *infile, *outfile, *ctfile, *tplan, buffer[80],
        holder[80];

    int gdriver = VGA, gmode = VGAHI, i, j, k, i1, i2, sli,
        pps, numin, jaws, num, errcode, lvs, cside, hit,
        *currxd, *curryd, *lastxd, *lastyd, plwid1, plwid2,
        lmove, oplf, *lfdis, arcs, arcnum, gpos, *ngp, srch1,
        srch2, plmid, m, m1, m2, hgt, hab, a, off, cx, cy,
        poly[8];

    long xmove, ymove, xlast, ylast, xunder, yunder;

    float scrndist = 67.1, viewdist = 100.0, ap1, ap2, la1,
        la2, ax1, ax2, del, st, rad, ap, lat, ax, ap0, la0,
        ax0, aprp, latrp, axrp, abrp, gtrp, ubrp, thk, gang,
        xgang, ogang, *tang, xtang, ggtmargin, gabmargin,

```

```

    otang, rinc, lwid, win, gsin, gcos, tsin, tcos,
    tcosgcos, tsingcos, tcosgsin, tsingsin, *abb, *gtb,
    *ubb, *abt, *gtt, *ubt, *abr, *gtr, *ubr, *lfpos,
    *gst, *gsp, *wt, gtmargin, abmargin, tran, angle,
    avrad, scale, cxcy, r2aop;

FILE *infp, *outfp, *ctfp;

rad = (M_PI) / 180.0;
r2aop = (float)sqrt(M_PI) / 4.;

infile = (char *)calloc(13, sizeof(char));
outfile = (char *)calloc(13, sizeof(char));
ctfile = (char *)calloc(13, sizeof(char));
tplan = (char *)calloc(13, sizeof(char));

if(argc != 4)
{
    printf("Multileaf collimator target localization\n\n");

    printf("Enter input file name: ");
    scanf("%s", infile);

    printf("Enter output file name: ");
    scanf("%s", outfile);

    printf("Enter number of jaws: ");
    scanf("%i", &jaws);
}
else
{
    infile = argv[1];
    outfile = argv[2];
    jaws = atoi(argv[3]);
}

// open input and output files
if((infp = fopen(infile, "r")) == NULL)
{
    printf("Input file %s not available\n", infile);
    exit(0);
}

if((outfp = fopen(outfile, "w")) == NULL)
{
    printf("Output file %s not available\n", outfile);
    exit(0);
}

clrscr();

// register and initialize graphics

```

```

if(registerbgidriver(EGAVGA_driver) < 0)
{
    printf("Graphics driver fault\n");
    exit(0);
}

initgraph(&gdriver, &gmode, "");

errcode = graphresult();
if(errcode != grOk)
{
    printf("Graphics error: %s\n", grapherrormsg(errcode));
    exit(0);
}

cleardevice();

setbkcolor(BLACK);

setviewport(0, 0, 639, 479, 1);

setcolor(WHITE);

setfillstyle(SOLID_FILL, BLUE);

settextjustify(LEFT_TEXT, CENTER_TEXT);

// read CT file name
fscanf(infp, "%s", ctfile);
if((ctfp = fopen(ctfile, "r")) == NULL)
{
    cleardevice();
    closegraph();
    clrscr();
    printf("CT file %s not available\n", ctfile);
    exit(0);
}

// read treatment plan
fscanf(infp, "%s", tplan);

// read number of arcs
fscanf(infp, "%i", &arcs);

tang = (float *)calloc((arcs + 1), sizeof(float));
gst = (float *)calloc((arcs + 1), sizeof(float));
gsp = (float *)calloc((arcs + 1), sizeof(float));
wt = (float *)calloc((arcs + 1), sizeof(float));
ngp = (int *)calloc((arcs + 1), sizeof(int));

// read rotation isocenter in BRW cs and convert to
// collimator cs

```

```

fscanf(infp, "%f %f %f", &aprp, &latrp, &axrp);
abrp = latrp;      // AB rotation point
gtrp = axrp;       // GT rotation point
ubrp = aprp;       // up beam rotation point

// read number of leaves, leaf width, and collimator
// transmission factor
fscanf(infp, "%i %f %f", &lvs, &lwid, &tran);
win = (float)lvs * lwid / 2.;
// leaf position (graphic)
lfdis = (int *)calloc((lvs * 4 + 1), sizeof(int));
// leaf position (real)
lfpos = (float *)calloc((lvs * 4 + 1), sizeof(float));

// read cutting point for dose calcs
fscanf(infp, "%f %f %f", &ap0, &la0, &ax0);

// read grid dim and spacing for dose calcs
fscanf(infp, "%f %f", &ap1, &ap2);
fscanf(infp, "%f %f", &la1, &la2);
fscanf(infp, "%f %f", &ax1, &ax2);
fscanf(infp, "%f", &del);

// read gantry rotation increment
fscanf(infp, "%f", &rinc);

// read treatment plan
for(j = 1; j <= arcs; j++)
{
    fscanf(infp, "%f %f %f %f", &tang[j], &gst[j], &gsp[j],
        &wt[j]);
    if(gst[j] == gsp[j])
    {
        gsp[j] = gst[j];
        ngp[j] = 1;
    }
    else
    {
        gst[j] = gst[j] + 0.5 * rinc;
        ngp[j] = (int)ceil((double)((gsp[j] - gst[j]) / rinc));
    }
}

// read collimator margin (cm)
fscanf(infp, "%f %f", &gtmargin, &abmargin);

ggtmargin = round(gtmargin * (401. / (2. * win)));
hgt = ggtmargin * ggtmargin;

gabmargin = round(abmargin * (401. / (2. * win)));
hab = gabmargin * gabmargin;

```

```

// read slice thickness, number of slices, points per slice
fscanf(ctfp, "%f", &thk);
fscanf(ctfp, "%i", &sli);
fscanf(ctfp, "%i", &pps);
numin = sli * pps;
num = 2 * numin;

// collimator AB axis base
abb = (float *)calloc((num + 1), sizeof(float));
// collimator GT axis base
gtb = (float *)calloc((num + 1), sizeof(float));
// collimator up beam base
ubb = (float *)calloc((num + 1), sizeof(float));
// AB rotated
abr = (float *)calloc((num + 1), sizeof(float));
// GT rotated
gtr = (float *)calloc((num + 1), sizeof(float));
// UB rotated
ubr = (float *)calloc((num + 1), sizeof(float));
// AB translated
abt = (float *)calloc((num + 1), sizeof(float));
// GT translated
gtt = (float *)calloc((num + 1), sizeof(float));
// UB translated
ubt = (float *)calloc((num + 1), sizeof(float));
// current x display
currxd = (int *)calloc((pps + 1), sizeof(int));
// current y display
curryd = (int *)calloc((pps + 1), sizeof(int));
// last x display
lastxd = (int *)calloc((pps + 1), sizeof(int));
// last y display
lastyd = (int *)calloc((pps + 1), sizeof(int));

// read the coordinates of the slice points and double for
// tiling
k = 0;
for(i = 1; i < numin; i += pps)
{
    for(j = 1; j <= pps; j++)
    {
        fscanf(ctfp, "%f %f %f", &ap, &lat, &ax);

        // calculate indices
        i1 = (i + j - 1) + (k * pps);
        i2 = (i + j - 1) + ((k + 1) * pps);

        // transform BRW coordinates to collimator coordinates
        abb[i1] = lat;
        abb[i2] = lat;
        gtb[i1] = ax + thk / 2.;
        gtb[i2] = ax - thk / 2.;
    }
}

```

```

    ubb[i1] = ap;
    ubb[i2] = ap;
}
k++;
}

fclose(infp);
free(infile);
fclose(ctfp);

// write treatment parameters
fprintf(outfp, "%s\n", ctfile);
fprintf(outfp, "%s\n", tplan);
fprintf(outfp, "%.2f %.2f\n", gtmargin, abmargin);
fprintf(outfp, "%i\n", arcs);
fprintf(outfp, "%.3f %.3f %.3f\n", aprp, latrp, axrp);
fprintf(outfp, "%i %.3f %.5f\n", lvs, lwid, tran);
fprintf(outfp, "%.3f\n", rinc);

for(i = 1; i <= arcs; i++)
{
    if(gst[i] == gsp[i])
        st = gst[i];
    else
        st = gst[i] - 0.5 * rinc;

    fprintf(outfp, "%.3f %.3f %.3f %.3f\n", tang[i], st,
        gsp[i], wt[i]);
}
fprintf(outfp, "%.3f %.3f %.3f\n", ap0, la0, ax0);

fprintf(outfp, "%.3f %.3f\n", ap1, ap2);
fprintf(outfp, "%.3f %.3f\n", la1, la2);
fprintf(outfp, "%.3f %.3f\n", ax1, ax2);
fprintf(outfp, "%.4f\n", del);

clearviewport();

// draw screen box
rectangle(0, 0, 639, 479);
floodfill(1, 1, WHITE);

// print view orientation
setcolor(GREEN);
outtextxy(317, 40, "G");
outtextxy(319, 463, "T");
outtextxy(107, 252, "A");
outtextxy(530, 252, "B");

// print working and CT file names
setcolor(YELLOW);
outtextxy(15, 15, strupr(outfile));

```



```

outtextxy(15, 30, strupr(ctfile));
free(ctfile);

setcolor(WHITE);

outtextxy(121, 15, "Gantry incr");

outtextxy(388, 15, "Arc");
strcpy(buffer, "of ");
strcat(buffer, itoa(arcs, holder, 10));
outtextxy(444, 15, buffer);

outtextxy(121, 30, "Gantry:");
outtextxy(241, 30, "deg");

outtextxy(388, 30, "Table:");
outtextxy(500, 30, "deg");

// draw viewport box
setcolor(YELLOW);
setviewport(119, 49, 523, 453, 1);
rectangle(0, 0, 404, 404);
rectangle(1, 1, 403, 403);

setfillstyle(SOLID_FILL, GREEN);

xgang = ogang = 0.;
xtang = otang = 0.;

for(arcnum = 1; arcnum <= arcs; arcnum++)
{
    // calculate table rotation sines and cosines
    otang = xtang;
    xtang = tang[arcnum];
    tsin = (float)sin((double)(-tang[arcnum] * rad));
    tcos = (float)cos((double)(-tang[arcnum] * rad));

    setviewport(0, 0, 639, 479, 1);

    // erase old arcs
    if(arcnum > 1)
    {
        setcolor(BLUE);
        setttextjustify(LEFT_TEXT, CENTER_TEXT);
        strcpy(buffer, "of ");
        strcat(buffer, itoa(ngp[arcnum - 1], holder, 10));
        outtextxy(241, 15, buffer);
        setttextjustify(RIGHT_TEXT, CENTER_TEXT);
        outtextxy(436, 15, itoa((arcnum - 1), buffer, 10));
        outtextxy(492, 30, ftoa(otang, 2));
    }
}

```

```

// print updated arcs
setcolor(WHITE);
settextjustify(LEFT_TEXT, CENTER_TEXT);
strcpy(buffer, "of ");
strcat(buffer, itoa(ngp[arcnum], holder, 10));
outtextxy(241, 15, buffer);
settextjustify(RIGHT_TEXT, CENTER_TEXT);
outtextxy(436, 15, itoa(arcnum, buffer, 10));
outtextxy(492, 30, ftoa(xtang, 2));

for(gpos = 1; gpos <= ngp[arcnum]; gpos++)
{
    // calculate gantry rotation sines and cosines plus
    // products
    gang = -(gst[arcnum] + ((float)gpos - 1.) * rinc);
    ogang = xgang;
    xgang = -gang;
    gsin = (float)sin((double)(gang * rad));
    gcos = (float)cos((double)(gang * rad));

    tcosgcos = tcos * gcos;
    tsingcos = tsin * gcos;
    tcosgsin = tcos * gsin;
    tsingsin = tsin * gsin;

    setviewport(0, 0, 639, 479, 1);

    // erase old gantry angles
    setcolor(BLUE);
    if(gpos == 1)
        outtextxy(233, 15, itoa(ngp[arcnum] - 1], buffer, 10));
    else
        outtextxy(233, 15, itoa((gpos - 1), buffer, 10));
    outtextxy(233, 30, ftoa(ogang, 2));

    // print updated gantry angles
    setcolor(WHITE);
    outtextxy(233, 15, itoa(gpos, buffer, 10));
    outtextxy(233, 30, ftoa(xgang, 2));

    setviewport(121, 51, 521, 451, 1);

    clearviewport();

    for(i = 1; i <= num; i++)
    {
        // translate rotation point to origin (BRW center to
        // isocenter)
        abt[i] = abb[i] - abrp;
        gtt[i] = gtb[i] - gtrp;
        ubt[i] = ubb[i] - ubrp;
    }
}

```

```

// rotate data about table then gantry
abr[i] = abt[i] * tcosgcos - gtt[i] * tsingcos +
        ubt[i] * gsin;
gtr[i] = abt[i] * tsin + gtt[i] * tcos;
ubr[i] = -abt[i] * tcosgsin + gtt[i] * tsingsin +
        ubt[i] * gcos;

// copy terms
abt[i] = abr[i];
gtt[i] = gtr[i];
ubt[i] = ubr[i];

// project 3D rotated points on 2D display plane
abt[i] = scrndist * abt[i] / (viewdist - ubt[i]);
gtt[i] = scrndist * gtt[i] / (viewdist - ubt[i]);
}

// draw rotated figure
k = 1;
for(i = 1; i < num; i += pps)
{
    for(j = 1; j <= pps; j++)
    {
        if(i > 1)
        {
            lastxd[j] = currxd[j];
            lastyd[j] = curryd[j];
        }
        currxd[j] = dispx(abt[i + j - 1], 401, win);
        curryd[j] = dispy(gtt[i + j - 1], 401, win);
    }
    // draw slice
    polydraw(currxd, curryd, pps);

    // tile if second slice in set
    if((k % 2) == 0)
        tile(currxd, curryd, lastxd, lastyd, pps);
    k++;
}

// find leaf positions
setcolor(GREEN);
j = 1;
plmid = round((lwid / 2.) * (401. / (2. * win)));
for(cside = 1; cside <= 4; cside++)
{
    switch(cside)
    {
        case 3:
            oplf = j - lvs - 1;
            break;
    }
}

```

```

case 4:
    oplf = j - lvs - 1;
break;

default:
break;
}

for(k = 1; k <= lvs; k++)
{
    hit = 0;
    switch(cside)
    {
        case 1: // gantry side
            lmove = -ggtmargin;

            if(jaws == 4)
            {
                plwid1 = dispx((-win + (k - 1) * lwid), 401, win);
                plwid2 = dispx((-win + k * lwid), 401, win);

                poly[0] = plwid1;
                poly[1] = lmove;
                poly[2] = plwid2;
                poly[3] = lmove;

                srch2 = plwid2 + gabmargin;
                if(srch2 > 400)
                    srch2 = 400;

                do
                {
                    lmove++;

                    putpixel(plwid1 + plmid, lmove, LIGHTRED);

                    if(k == 1)
                        srch1 = plwid1;
                    else
                    {
                        if(lmove <= lfdis[j - 1])
                            srch1 = plwid1;
                        else
                        {
                            srch1 = plwid1 - gabmargin;
                            if(srch1 < 0)
                                srch1 = 0;
                        }
                    }
                }

                i = plwid1;
                do

```

```

{
  m = limit(lmove + ggtmargin, 0, 400);

  if(getpixel(i, m) == WHITE)
    hit = 1;

  i += 2;
}while(!hit && (i <= plwid2));

if((!hit) && (abmargin != 0.))
{
  i = plwid2 + 1;
  off = 1;
  do
  {
    a = off * off;
    m2 = round((float)sqrt((double)(hab - a)));
    if(m2 > ggtmargin)
      m2 = ggtmargin;
    m = limit(lmove + m2, 0, 400);

    if(getpixel(i, m) == WHITE)
      hit = 1;

    i += 2;
    off += 2;
  }while(!hit && (i <= srch2));

  i = plwid1 - 1;
  off = 1;
  do
  {
    a = off * off;
    m1 = round((float)sqrt((double)(hab - a)));
    if(m1 > ggtmargin)
      m1 = ggtmargin;
    m = limit(lmove + m1, 0, 400);

    if(getpixel(i, m) == WHITE)
      hit = 1;

    i -= 2;
    off += 2;
  }while(!hit && (i >= srch1));
}
}while(!hit && (lmove <= 400));
}

lmove = limit(lmove, 0, 401);

lfdis[j] = lmove;

```

```

if(jaws == 4)
{
    poly[4] = plwid2;
    poly[5] = lmove;
    poly[6] = plwid1;
    poly[7] = lmove;
    fillpoly(4, poly);
}

lfpos[j] = realy(lmove, 401, win);

j++;
break;

case 2: // B side
    lmove = 400 + gabmargin;

    plwid1 = dispy((win - (k - 1) * lwid), 401, win);
    plwid2 = dispy((win - k * lwid), 401, win);

    poly[0] = lmove;
    poly[1] = plwid1;
    poly[2] = lmove;
    poly[3] = plwid2;

    srch2 = plwid2 + ggtmargin;
    if(srch2 > 400)
        srch2 = 400;

    do
    {
        lmove--;

        putpixel(lmove, plwid1 + plmid, LIGHTRED);

        if(k == 1)
            srch1 = plwid1;
        else
        {
            if(lmove >= lfdis[j - 1])
                srch1 = plwid1;
            else
            {
                srch1 = plwid1 - ggtmargin;
                if(srch1 < 1)
                    srch1 = 1;
            }
        }
    }

    i = plwid1;
    do
    {

```

```

m = limit(lmove - gabmargin, 0, 400);

if(getpixel(m, i) == WHITE)
    hit = 1;

i += 2;
}while(!hit && (i <= plwid2));

if((!hit) && (gtmargin != 0.))
{
    i = plwid2 + 1;
    off = 1;
    do
    {
        a = off * off;
        m2 = round((float)sqrt((double)(hgt - a)));
        if(m2 > gabmargin)
            m2 = gabmargin;
        m = limit(lmove - m2, 0, 400);

        if(getpixel(m, i) == WHITE)
            hit = 1;

        i += 2;
        off += 2;
    }while(!hit && (i <= srch2));

    i = plwid1 - 1;
    off = 1;
    do
    {
        a = off * off;
        m1 = round((float)sqrt((double)(hgt - a)));
        if(m1 > gabmargin)
            m1 = gabmargin;
        m = limit(lmove - m1, 0, 400);

        if(getpixel(m, i) == WHITE)
            hit = 1;

        i -= 2;
        off += 2;
    }while(!hit && (i >= srch1));
}
}while(!hit && (lmove >= 0));

lmove = limit(lmove, 0, 401);

lfdis[j] = lmove;

poly[4] = lmove;
poly[5] = plwid2;

```

```

poly[6] = lmove;
poly[7] = plwid1;
fillpoly(4, poly);

lfpos[j] = realx(lmove, 401, win);

j++;
break;

case 3: // table side
    lmove = 400 + ggtmargin;

    if(jaws == 4)
    {
        plwid1 = dispX((win - k * lwid), 401, win);
        plwid2 = dispX((win - (k - 1) * lwid), 401, win);

        poly[0] = plwid1;
        poly[1] = lmove;
        poly[2] = plwid2;
        poly[3] = lmove;

        srch1 = plwid1 - gabmargin;
        if(srch1 < 1)
            srch1 = 1;

        do
        {
            if(lmove <= lfdis[oplf])
            {
                lmove = lfdis[oplf];
                hit = 1;
            }
            else
            {
                lmove--;

                putpixel(plwid1 + plmid, lmove, LIGHTRED);

                if(k == 1)
                    srch2 = plwid2;
                else
                {
                    if(lmove >= lfdis[j - 1])
                        srch2 = plwid2;
                    else
                    {
                        srch2 = plwid2 + gabmargin;
                        if(srch2 > 400)
                            srch2 = 400;
                    }
                }
            }
        }
    }

```



```

i = plwid1;
do
{
    m = limit(lmove ~ ggtmargin, 0, 400);

    if(getpixel(i, m) == WHITE)
        hit = 1;

    i += 2;
}while(!hit && (i <= plwid2));

if((!hit) && (abmargin != 0))
{
    i = plwid2 + 1;
    off = 1;
    do
    {
        a = off * off;
        m2 = round((float)sqrt((double)(hab - a)));
        if(m2 > ggtmargin)
            m2 = ggtmargin;
        m = limit(lmove - m2, 0, 400);

        if(getpixel(i, m) == WHITE)
            hit = 1;

        i += 2;
        off += 2;
    }while(!hit && (i <= srch2));

    i = plwid1 - 1;
    off = 1;
    do
    {
        a = off * off;
        m1 = round((float)sqrt((double)(hab - a)));
        if(m1 > ggtmargin)
            m1 = ggtmargin;
        m = limit(lmove - m1, 0, 400);

        if(getpixel(i, m) == WHITE)
            hit = 1;

        i -= 2;
        off += 2;
    }while(!hit && (i >= srch1));
    }
}while(!hit && (lmove >= 0));
}

lmove = limit(lmove, 0, 401);

```

```

lfdis[j] = lmove;

if(jaws == 4)
{
    poly[4] = plwid2;
    poly[5] = lmove;
    poly[6] = plwid1;
    poly[7] = lmove;
    fillpoly(4, poly);
}

lfpos[j] = realy(lmove, 401, win);

oplf--;
j++;
break;

case 4: // A side
    lmove = -gabmargin;

    plwid1 = dispy((-win + k * lwid), 401, win);
    plwid2 = dispy((-win + (k - 1) * lwid), 401, win);

    poly[0] = lmove;
    poly[1] = plwid1;
    poly[2] = lmove;
    poly[3] = plwid2;

    srch1 = plwid1 - ggtmargin;
    if(srch1 < 1)
        srch1 = 1;

do
{
    if(lmove >= lfdis[oplf])
    {
        lmove = lfdis[oplf];
        hit = 1;
    }
    else
    {
        lmove++;

        putpixel(lmove, plwid1 + plmid, LIGHTRED);

        if(k == 1)
            srch2 = plwid2;
        else
        {
            if(lmove <= lfdis[j - 1])
                srch2 = plwid2;
            else

```

```

{
    srch2 = plwid2 + ggtmargin;
    if(srch2 > 400)
        srch2 = 400;
}
}

i = plwid1;
do
{
    m = limit(lmove + gabmargin, 0, 400);

    if(getpixel(m, i) == WHITE)
        hit = 1;

    i += 2;
}while(!hit && (i <= plwid2));

if(!hit && (gtmargin != 0.))
{
    i = plwid2 + 1;
    off = 1;
    do
    {
        a = off * off;
        m2 = round((float)sqrt((double)(hgt - a)));
        if(m2 > gabmargin)
            m2 = gabmargin;
        m = limit(lmove + m2, 0, 400);

        if(getpixel(m, i) == WHITE)
            hit = 1;

        i += 2;
        off += 2;
    }while(!hit && (i <= srch2));

    i = plwid1 - 1;
    off = 1;
    do
    {
        a = off * off;
        m1 = round((float)sqrt((double)(hgt - a)));
        if(m1 > gabmargin)
            m1 = gabmargin;
        m = limit(lmove + m1, 0, 400);

        if(getpixel(m, i) == WHITE)
            hit = 1;

        i -= 2;
        off += 2;
    }
}

```

```

        }while(!hit && (i >= srchl));
    }
}while(!hit && (lmove <= 400));

lmove = limit(lmove, 0, 401);

lfdis[j] = lmove;

poly[4] = lmove;
poly[5] = plwid2;
poly[6] = lmove;
poly[7] = plwid1;
fillpoly(4, poly);

lfpos[j] = realx(lmove, 401, win);

    oplf--;
    j++;
    break;
}
} // next leaf
} // next cside

// find average collimator radius
scale = 2. * win / 401.;
avrad = 0.;
for(angle = 0.; angle < 360.; angle += 5.)
{
    j = 1;
    hit = 0;
    cx = 200;
    cy = 200;
    xlast = xunder = 0;
    ylast = yunder = 0;

    do
    {
        if(getpixel(cx, cy) == YELLOW)
            hit = 1;

        if(!hit)
        {
            xmove = round((float)j * (float)sin((double)(angle *
                rad)));
            ymove = round((float)j * (float)cos((double)(angle *
                rad)));

            cx = 200 + (int)xmove;
            cy = 200 + (int)ymove;

            if(getpixel(cx, cy) != GREEN)

```

```

        putpixel(cx, cy, LIGHTCYAN);
    else
    {
        if(xmove != xlast)
            xunder++;

        if(ymove != ylast)
            yunder++;
    }

    j++;

    xlast = xmove;
    ylast = ymove;
}
}while(!hit);

xmove = labs(xmove) - xunder;
ymove = labs(ymove) - yunder;

cxcy = (float)xmove * (float)xmove + (float)ymove *
        (float)ymove;
cxcy = (float)sqrt((double)cxcy);

avrad += cxcy * scale / 72.;
}

// convert average radius to area/perimeter
avrad *= r2aop;

fprintf(outfp, "%.4f\n", avrad);

// print leaf positions
for(j = 1; j <= (lvs * 4); j++)
    fprintf(outfp, "%.2f ", lfpos[j]);
fprintf(outfp, "\n");

} // next gpos
} // next arcnum

fclose(outfp);

cleardevice();

closegraph();

clrscr();

free(outfile);
free(tplan);
free(tang);
free(gst);

```

```

free(gsp);
free(wt);
free(ngp);
free(abb);
free(gtb);
free(ubb);
free(abr);
free(gtr);
free(ubr);
free(abt);
free(gtt);
free(ubt);
free(currxd);
free(curryd);
free(lastxd);
free(lastyd);
free(lfpos);
free(lfdis);
}

// draw a polygon
void polydraw(int *x, int *y, int pts)
{
    int i;

    moveto(x[1], y[1]);

    for(i = 2; i <= pts; i++)
        lineto(x[i], y[i]);

    lineto(x[1], y[1]);
}

// connect two polygons at corresponding points
void tile(int *x1, int *y1, int *x0, int *y0, int pts)
{
    int i;

    for(i = 1; i <= pts; i++)
    {
        moveto(x1[i], y1[i]);
        lineto(x0[i], y0[i]);
    }
}

// calculate x axis screen display point
int dispx(float xpt, int wide, float window)
{
    int out;
    float scale, posit;

    scale = wide / (2 * window);

```

```

posit = (wide / 2) + (xpt * scale);
out = round(posit);

return out;
}

// calculate real space point from x axis screen display
float realx(int posit, int wide, float window)
{
    float scale, out;

    scale = wide / (2 * window);

    out = (posit - (wide / 2.)) / scale;

    return out;
}

// calculate y axis screen display point
int dispy(float ypt, int high, float window)
{
    int out;
    float scale, posit;

    scale = high / (2 * window);

    posit = (high / 2) - (ypt * scale);

    out = round(posit);

    return out;
}

// calculate real space point from y axis screen display
float realy(int posit, int high, float window)
{
    float scale, out;

    scale = high / (2 * window);

    out = ((high / 2.) - posit) / scale;

    return out;
}

// round floating point input to integer
int round(float in)
{
    int out;

    out = (int)floor((double)in + 0.5);
}

```

```

    return out;
}

// calculate precision necessary for ftoa()
int prec(double index, int pl)
{
    int p;

    if(fabs(index) < 1.)
        p = pl;
    else
        p = (int)floor(log10(fabs(index))) + pl;

    return p;
}

// convert floating point number to character string and
// store in fstr
char *ftoa(float fin, int place)
{
    int i, dec, sign;
    char *numstr;
    static char fstr[10];

    numstr = (char *)calloc(10, sizeof(char));

    strcpy(fstr, "");

    // convert input floating point to string
    numstr = fcvt((double)fin, prec((double)fin, place), &dec,
                  &sign);

    // append minus if negative
    if((sign != 0) && (strlen(numstr) != 0))
        strcat(fstr, "-");

    // if no whole number part
    if(dec <= 0)
    {
        strcat(fstr, "0.");
        if((strlen(numstr) == 0) || (abs(dec) >= place))
            for(i = 1; i <= place; i++)
                strcat(fstr, "0");
        else
        {
            for(i = 1; i <= abs(dec); i++)
                strcat(fstr, "0");
            strcat(fstr, numstr);
        }
    }
    // if whole number plus decimal
    else

```



```
{
    strncat(fstr, numstr, dec);
    strcat(fstr, "." );
    strncat(fstr, numstr + dec, place);
}

strcat(fstr, "\0");

free(numstr);

return(fstr);
}

// limit input
int limit(int in, int lo, int hi)
{
    int out;

    if(in < lo)
        in = lo;

    if(in > hi)
        in = hi;

    out = in;

    return out;
}
```

## APPENDIX C DOSE MODEL PROGRAM (PLANE)

The dose model program, LFDUSE.C, computes plane dosimetry in the axial, sagittal, or coronal planes. This process is illustrated in figure C-1. The command line takes arguments for the input file (from LFLOC.C), the name of the output file, a single letter (A for axial, S for sagittal, or C for coronal) for the desired plane of calculation, and a number for the desired type of matrix normalization (0 results in no normalization, -1 normalizes to plane maximum, 1 normalizes to isocenter, and any other positive floating point value normalizes to that value).

The output file header is written in the EasyPlot<sup>™</sup> (Spiral Software, Brookline MA) batch language for export to that program to result in generation of contour or cross plots. Immediately succeeding the batch lines, the output data is written in the form of a two dimensional matrix that directly represents the dosimetry plane matrix for the contour plots or in an xy list for cross plots.

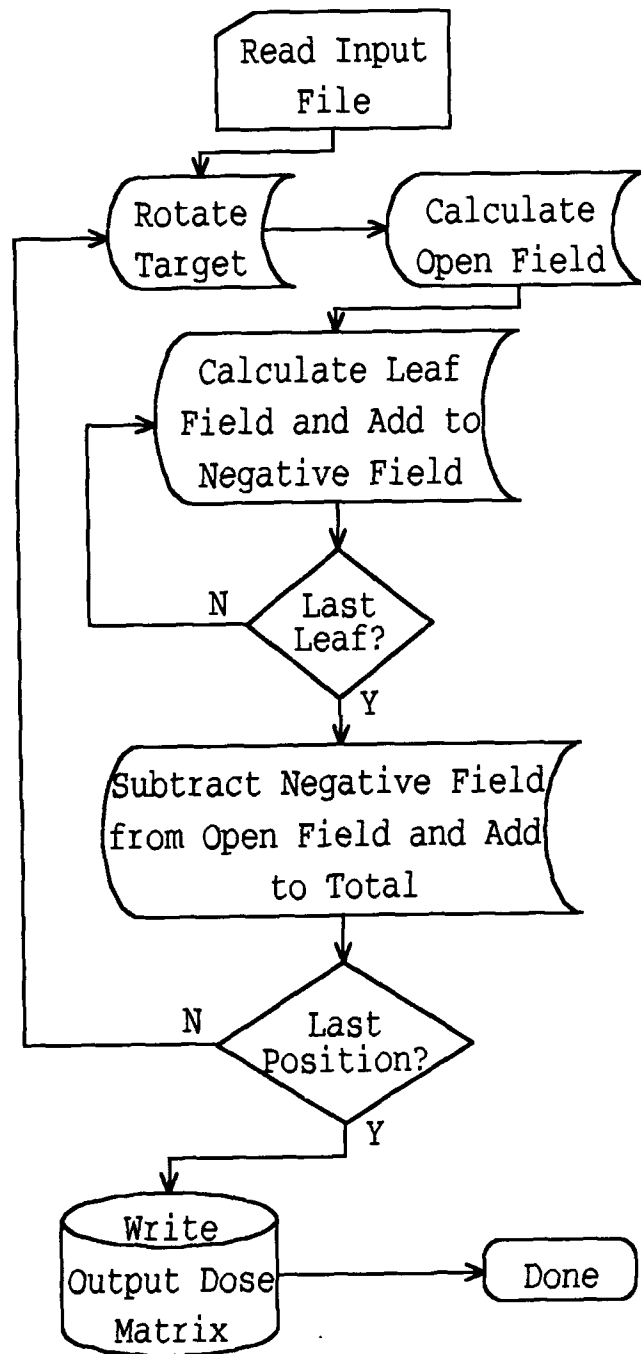


Figure C-1: LFDOSE.C flow chart

```

// Program LFDOSE.C
//   This program calculates dose in 3D space using the
// modified negative field method with a multileaf
// collimator
//   It can calculate dose on any of three orthogonal planes
// (axial, sagittal, or coronal)
//   Takes arguments interactively or on the command line
//   Command line LFDOSE <input file> <output file> <plane>
// <norm> <dose>
//   Input file is generated by LFLOC.C
//   Output file is setup for batch file input to Easy Plot
// for contour generation
//   Plane is A (axial), S (sagittal), or C (coronal)
//   Norm is 0 (no normalization), -1 (norm to plane max), 1
// (norm to isocenter), or any previously determined
// floating point value
//   Dose is set monitor units
//
// Notes:
//   Polynomial fitting for TMR within build-up region
//   Exponential fitting for TMR in equilibrium region
//   Modified Cunningham model for OAR
//   Dose set to zero outside the phantom
//   Beam data = 6 MVX LINAC at Shands Cancer Center
//
// Based on DOSEEXRB.FOR written by Tae-Suk Suh [Suh90]

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <math.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>

#define BACKSPACE 8

float pdose(float fa, float fb, float fg, float ft,
            float of, float aop);
float **matrix(int rl, int rh, int cl, int ch);
void free_matrix(float **m, int rl, int rh, int cl);
int *ivector(int l, int h);
void free_ivector(int *v, int l);
float *vector(int l, int h);
void free_vector(float *v, int l);

// constant values
//   sad = source to axis distance = 100 cm
//   scd = source to collimator distance = 67.1 cm
//   ss = source size = 0.2 cm
//   dmax = 1.5 cm for 6MVX
static float sad = 100.0,

```

```

        scd = 67.1,
        ss = 0.2,
        dmax = 1.5;

// global variables
static float ap, la, ax, api, lai, axi, disoij, di, r0, r2,
            tran, gcos, gsin, tcos, tsin, bfa, bfb, bfg,
            bft;

void main(int argc, char *argv[])
{
    int i, j, k, arcs, lvs, *ngp, dlx, dly, dax, dla, dap,
        cside, nocalc, matmin, spinpos, gptot;

    float rad, lwid, ginc, ap0, la0, ax0, ap1, ap2, la1, la2,
        ax1, ax2, temp, del, ri2, valc, gang, b, bbcc, lsa,
        lsb, lsg, lst, *wt, maxdose, isomax, *tang, *gsp,
        *gst, gtmargin, abmargin, *leaf, **negf, **dout, cut,
        dginc, pt, of, mag, caop, rxdose;

    char *infile, *outfile, *ctfile, *tplan, iplane,
        buffer[80], spinner[] = {"\\|/-"};

    FILE *infp, *outfp;

    rad = (M_PI) / 180.;

    infile = (char *)calloc(14, sizeof(char));
    outfile = (char *)calloc(14, sizeof(char));
    ctfile = (char *)calloc(14, sizeof(char));
    tplan = (char *)calloc(14, sizeof(char));

    printf("\nMultileaf collimator dosimetry for principal
        planes\n");

    if(argc != 6)
    {
        printf("Enter input file name: ");
        scanf("%s", infile);
        infile = strupr(infile);

        printf("Enter output file name: ");
        scanf("%s", outfile);
        outfile = strupr(outfile);

        do
        {
            printf("Enter dose matrix plane - <A>xial, <S>agittal,
                <C>oronal: ");
            iplane = getche();
            iplane = toupper(iplane);
            printf("\n");

```

```

}while((iplane != 'A') && (iplane != 'S') &&
      (iplane != 'C')));

printf("Enter matrix max dose:\n");
printf("(0 => no norm, -1 => norm to plane max, 1 => norm
      to isocenter:) ");
scanf("%f", &maxdose);

printf("Enter monitor units applied (cGy): ");
scanf("%f", &rxdose);
}
else
{
  infile = strupr(argv[1]);
  outfile = strupr(argv[2]);
  strncpy(&iplane, argv[3], 1);
  iplane = toupper(iplane);
  maxdose = (float)atof(argv[4]);
  rxdose = (float)atof(argv[5]);
}

_setcursortype(_NOCURSOR);

if((infp = fopen(infile, "r")) == NULL)
{
  printf("Input file %s unavailable.\n", infile);
  exit(0);
}

if((outfp = fopen(outfile, "w")) == NULL)
{
  printf("Cannot open %s for output.\n", outfile);
  exit(0);
}

fscanf(infp, "%s", ctfile);
ctfile = strupr(ctfile);

fscanf(infp, "%s", tplan);
tplan = strupr(tplan);

fscanf(infp, "%f %f", &gtmargin, &abmargin);

printf("Working on %s for target file %s ...\n", outfile,
      ctfile);

// read treatment plan
fscanf(infp, "%i", &arcs);

fscanf(infp, "%f %f %f", &api, &lai, &axi);

fscanf(infp, "%i", &lvs);

```

```

fscanf(infp, "%f", &lwid);
bfb = bfg = lvs * lwid / 2.;
bfa = bft = -lvs * lwid / 2.;

fscanf(infp, "%f", &tran);

fscanf(infp, "%f", &ginc);
dginc = ginc;
ginc *= rad;

tang = vector(0, arcs); // table angle
gst = vector(0, arcs); // gantry start angle
gsp = vector(0, arcs); // gantry stop angle
wt = vector(0, arcs); // arc weight
ngp = ivector(0, arcs); // number of gantry positions
leaf = vector(0, lvs); // leaf positions per side

gptot = 0;
for(j = 1; j <= arcs; j++)
{
    fscanf(infp, "%f %f %f %f", &tang[j], &gst[j], &gsp[j],
        &wt[j]);
    tang[j] *= rad;
    if(gst[j] == gsp[j])
    {
        gst[j] *= rad;
        gsp[j] = gst[j];
        ngp[j] = 1;
        gptot = 1;
    }
    else
    {
        gst[j] = gst[j] * rad + 0.5 * ginc;
        gsp[j] *= rad;
        ngp[j] = (int)ceil((double)((gsp[j] - gst[j]) / ginc));
        gptot += ngp[j];
    }
}

// read dose grid center
fscanf(infp, "%f %f %f", &ap0, &la0, &ax0);

// read dose grid size and spacing
fscanf(infp, "%f %f", &ap1, &ap2);
fscanf(infp, "%f %f", &la1, &la2);
fscanf(infp, "%f %f", &ax1, &ax2);
fscanf(infp, "%f", &del);

switch(iplane)
{
    case 'A':
        if(la1 > la2)

```

```

{
    temp = la1;
    la1 = la2;
    la2 = temp;
}
dlx = 1 + (int)ceil((double)((la2 - la1) / del));

if(ap1 > ap2)
{
    temp = ap1;
    ap1 = ap2;
    ap2 = temp;
}
dly = 1 + (int)ceil((double)((ap2 - ap1) / del));

printf(" %i lateral points, %i AP points\n", dlx, dly);
break;

case 'S':
    if(ax1 > ax2)
    {
        temp = ax1;
        ax1 = ax2;
        ax2 = temp;
    }
    dlx = 1 + (int)ceil((double)((ax2 - ax1) / del));

    if(ap1 > ap2)
    {
        temp = ap1;
        ap1 = ap2;
        ap2 = temp;
    }
    dly = 1 + (int)ceil((double)((ap2 - ap1) / del));

    printf(" %i axial points, %i AP points\n", dlx, dly);
break;

case 'C':
    if(la1 > la2)
    {
        temp = la1;
        la1 = la2;
        la2 = temp;
    }
    dlx = 1 + (int)ceil((double)((la2 - la1) / del));

    if(ax1 > ax2)
    {
        temp = ax1;
        ax1 = ax2;
        ax2 = temp;
    }

```



```

    }
    dly = 1 + (int)ceil((double)((ax2 - ax1) / del));

    printf(" %i lateral points, %i axial points\n", dlx,
           dly);
    break;
}

if((dlx == 1) || (dly == 1))
    matmin = 0;
else
    matmin = 1;
// negative field matrix
negf = matrix(matmin, dly, matmin, dlx);
// output dose matrix
dout = matrix(matmin, dly, matmin, dlx);

// begin dose calculation

if(maxdose == 1.)
    isomax = 0.;

// patient model di cm diameter sphere
di = 20.;
r0 = 0.5 * di;
r2 = 0.25 * di * di;

// calculate point doses on grid
spinpos = 0;
for(j = 1; j <= arcs; j++)
{
    printf("\rCalculating arc %i of %i ", j, arcs);

    // parameters used to calculate depth of isocenter, disoi
    ri2 = api * api + lai * lai + axi * axi;
    valc = r2 - ri2;

    // calculate dose for gantry angle i (for arc j)
    for(i = 1; i <= ngp[j]; i++)
    {
        gang = gst[j] + ((float)i - 1.) * ginc;
        gcos = (float)cos((double)gang);
        gsin = (float)sin((double)gang);
        tcos = (float)cos((double)tang[j]);
        tsin = (float)sin((double)tang[j]);

        // calculate depth of isocenter for gantry i and arc j
        b = api * gcos + lai * gsin * tcos + axi * gsin * tsin;
        bbcc = b * b + valc;
        disoi = -b + (float)sqrt((double)bbcc);

        // read collimator area/perimeter

```

```

fscanf(infp, "%f", &caop);

// determine output factor for collimator
mag = sad / scd;
of = 1. - 0.4371 * exp(-2.627 * caop * mag);

// calculate point doses for base field
switch(iplane)
{
case 'A':
    ax = cut = ax0;
    ap = ap2;
    for(dap = 1; dap <= dly; dap++)
    {
        la = la1;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
        spinpos &= 0x03;
        for(dla = 1; dla <= dlx; dla++)
        {
            pt = pdose(bfa, bfb, bfg, bft, of, caop) * wt[j];
            dout[dap][dla] += pt;
            la += del;
        }
        ap -= del;
    }
    break;

case 'S':
    la = cut = la0;
    ap = ap2;
    for(dap = 1; dap <= dly; dap++)
    {
        ax = ax1;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
        spinpos &= 0x03;
        for(dax = 1; dax <= dlx; dax++)
        {
            pt = pdose(bfa, bfb, bfg, bft, of, caop) * wt[j];
            dout[dap][dax] += pt;
            ax += del;
        }
        ap -= del;
    }
    break;

case 'C':
    ap = cut = ap0;
    ax = ax2;
    for(dax = 1; dax <= dly; dax++)
    {
        la = la1;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
    }
}

```

```

spinpos &= 0x03;
for(dla = 1; dla <= dlx; dla++)
{
    pt = pdose(bfa, bfb, bfg, bft, of, caop) * wt[j];
    dout[dax][dla] += pt;
    la += del;
}
ax -= del;
}
break;
}

// find base isocenter max if requested
if(maxdose == 1.)
{
    ax = axi;
    la = lai;
    ap = api;
    pt = pdose(bfa, bfb, bfg, bft, of, caop) * wt[j];
    isomax += pt;
}

// calculate negative field doses for each leaf
for(cside = 1; cside <= 4; cside++)
{
    for(k = 1; k <= lvs; k++)
        fscanf(infp, "%f", &leaf[k]);

    for(k = 1; k <= lvs; k++)
    {
        nocalc = 0;

        switch(cside)
        {
            // gantry side
            case 1:
                if(leaf[k] >= bfg)
                    nocalc = 1;
                else
                {
                    lsa = bfa + (k - 1) * lwid;
                    lsb = lsa + lwid;
                    lsg = bfg;
                    lst = leaf[k];
                }
                break;

            // B side
            case 2:
                if(leaf[k] >= bfb)
                    nocalc = 1;
                else

```

```

    {
        lsa = leaf[k];
        lsb = bfb;
        lsg = bfg - (k - 1) * lwid;
        lst = lsg - lwid;
    }
    break;

// table side
case 3:
    if(leaf[k] <= bft)
        nocalc = 1;
    else
    {
        lsa = bfb - k * lwid;
        lsb = lsa + lwid;
        lsg = leaf[k];
        lst = bft;
    }
    break;

// A side
case 4:
    if(leaf[k] <= bfa)
        nocalc = 1;
    else
    {
        lsa = bfa;
        lsb = leaf[k];
        lsg = bft + k * lwid;
        lst = lsg - lwid;
    }
    break;
}

// calculate negative field if leaf in open base field
if(!nocalc)
{
    // calculate point doses for negative field
    switch(iplane)
    {
        case 'A':
            ax = ax0;
            ap = ap2;
            for(dap = 1; dap <= dly; dap++)
            {
                la = la1;
                printf("%c%c", spinner[spinpos++], BACKSPACE);
                spinpos &= 0x03;
                for(dla = 1; dla <= dlx; dla++)
                {
                    pt = pdose(lsa, lsb, lsg, lst, of, caop) * wt[j] *

```

```

        (1. - tran);
        negf[dap][dla] += pt;
        la += del;
    }
    ap -= del;
}
break;

case 'S':
    la = la0;
    ap = ap2;
    for(dap = 1; dap <= dly; dap++)
    {
        ax = ax1;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
        spinpos &= 0x03;
        for(dax = 1; dax <= dlx; dax++)
        {
            pt = pdose(lsa, lsb, lsg, lst, of, caop) * wt[j] *
                (1. - tran);
            negf[dap][dax] += pt;
            ax += del;
        }
        ap -= del;
    }
break;

case 'C':
    ap = ap0;
    ax = ax2;
    for(dax = 1; dax <= dly; dax++)
    {
        la = la1;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
        spinpos &= 0x03;
        for(dla = 1; dla <= dlx; dla++)
        {
            pt = pdose(lsa, lsb, lsg, lst, of, caop) * wt[j] *
                (1. - tran);
            negf[dax][dla] += pt;
            la += del;
        }
        ax -= del;
    }
break;
} // end switch

// find leaf isocenter max if requested
if(maxdose == 1.)
{
    ax = axi;
    la = lai;

```

```

        ap = api;
        pt = pdose(lsa, lsb, lsg, lst, of, caop) * wt[j] *
            (1. - tran);
        isomax -= pt;
    }
    } // end calc
    } // next leaf
    } // next collimator side
    } // next gantry angle
} // next arc

// subtract negative from base field and apply rx dose
for(i = 1; i <= dly; i++)
    for(j = 1; j <= dlx; j++)
    {
        dout[i][j] -= negf[i][j];

        if(dout[i][j] < tran)
            dout[i][j] = tran;

        dout[i][j] *= rxdose / (float)gptot;
    }

// normalize if asked
if(maxdose == 0.)
    strcpy(buffer, "no normalization");
if(maxdose == -1.)
    strcpy(buffer, "norm to plane max");
if(maxdose == 1.)
    strcpy(buffer, "norm to isocenter");
if(maxdose > 1.)
    strcpy(buffer, "norm to DVH max");

if(maxdose != 0.)
{
    if(maxdose == -1.)
        for(i = 1; i <= dly; i++)
            for(j = 1; j <= dlx; j++)
                if(dout[i][j] > maxdose)
                    maxdose = dout[i][j];

    if(maxdose == 1.)
        maxdose = isomax * rxdose / (float)gptot;

    for(i = 1; i <= dly; i++)
        for(j = 1; j <= dlx; j++)
            dout[i][j] /= maxdose;
}

// print Easy Plot batch output file
switch(iplane)
{

```

```

case 'A':
  if(la1 == la2)
  {
    fprintf(outfp, "/sm ON\n");
    fprintf(outfp, "/td xy\n");
    fprintf(outfp, "/et g 'Axial (%.3f cm), %s'\n", cut,
              ctfile);
    fprintf(outfp, "//gt2 'AP cross plot (%.2f cm
                    lateral)'\n", la1);
    fprintf(outfp, "/et g 'AP Cross Plot'\n");
    fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
    if(maxdose != 0.)
      fprintf(outfp, "/et y 'Normalized Dose'\n");
    else
      fprintf(outfp, "/et y 'Dose (cGy)'\n");
  }
  else if(ap1 == ap2)
  {
    fprintf(outfp, "/sm ON\n");
    fprintf(outfp, "/td xy\n");
    fprintf(outfp, "/et g 'Axial (%.3f cm), %s'\n", cut,
              ctfile);
    fprintf(outfp, "//gt2 'Lateral cross plot (%.2f cm
                    AP)'\n", ap1);
    fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
    if(maxdose != 0.)
      fprintf(outfp, "/et y 'Normalized Dose'\n");
    else
      fprintf(outfp, "/et y 'Dose (cGy)'\n");
  }
  else
  {
    fprintf(outfp, "//pos 0 0 0.76 1\n");
    fprintf(outfp, "/sm OFF\n");
    fprintf(outfp, "/td z\n");
    fprintf(outfp, "//cx %.2f %.2f\n", la1, la2);
    fprintf(outfp, "//cy %.2f %.2f\n", ap1, ap2);
    fprintf(outfp, "/et g 'Axial (%.3f cm), %s, %s'\n", cut,
              ctfile, tplan);
    fprintf(outfp, "//gt2 '%i leaves @ %.2f cm width, %.0f
                    deg rot incr'\n", lvs, lwid, dginc);
    fprintf(outfp, "//gt3 '%.2f cm GT %.2f cm AB margin,
                    %s'\n", gtmargin, abmargin, buffer);
    fprintf(outfp, "/et x 'R <- Lateral (cm) -> L'\n");
    fprintf(outfp, "/et y 'P <- AP (cm) -> A'\n");
  }
  break;

case 'S':
  if(ax1 == ax2)
  {
    fprintf(outfp, "/sm ON\n");

```

```

fprintf(outfp, "/td xy\n");
fprintf(outfp, "/et g 'Sagittal (%.3f cm), %s'\n", cut,
        ctfile);
fprintf(outfp, "//gt2 'AP cross plot (%.2f cm
        axial)\n", ax1);
fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
if(maxdose != 0.)
    fprintf(outfp, "/et y 'Normalized Dose'\n");
else
    fprintf(outfp, "/et y 'Dose (cGy)'\n");
}
else if(ap1 == ap2)
{
    fprintf(outfp, "/sm ON\n");
    fprintf(outfp, "/td xy\n");
    fprintf(outfp, "/et g 'Sagittal (%.3f cm), %s'\n", cut,
            ctfile);
    fprintf(outfp, "//gt2 'Axial cross plot (%.2f cm
            AP)\n", ap1);
    fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
    if(maxdose != 0.)
        fprintf(outfp, "/et y 'Normalized Dose'\n");
    else
        fprintf(outfp, "/et y 'Dose (cGy)'\n");
}
else
{
    fprintf(outfp, "//pos 0 0 0.76 1\n");
    fprintf(outfp, "/sm OFF\n");
    fprintf(outfp, "/td z\n");
    fprintf(outfp, "//cx %.2f %.2f\n", ax1, ax2);
    fprintf(outfp, "//cy %.2f %.2f\n", ap1, ap2);
    fprintf(outfp, "/et g 'Sagittal (%.3f cm), %s, %s'\n",
            cut, ctfile, tplan);
    fprintf(outfp, "//gt2 '%i leaves @ %.2f cm width, %.0f
            deg rot incr'\n", lvs, lwid, dginc);
    fprintf(outfp, "//gt3 '%.2f cm GT %.2f cm AB margin,
            %s'\n", gtmargin, abmargin, buffer);
    fprintf(outfp, "/et x 'Inf <- Axial (cm) -> Sup'\n");
    fprintf(outfp, "/et y 'P <- AP (cm) -> A'\n");
}
break;

case 'C':
    if(la1 == la2)
    {
        fprintf(outfp, "/sm ON\n");
        fprintf(outfp, "/td xy\n");
        fprintf(outfp, "/et g 'Coronal (%.3f cm), %s'\n", cut,
                ctfile);
        fprintf(outfp, "//gt2 'Axial cross plot (%.2f cm
                lateral)\n", la1);
    }

```



```

    fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
    if(maxdose != 0.)
        fprintf(outfp, "/et y 'Normalized Dose'\n");
    else
        fprintf(outfp, "/et y 'Dose (cGy)'\n");
}
else if(ax1 == ax2)
{
    fprintf(outfp, "/sm ON\n");
    fprintf(outfp, "/td xy\n");
    fprintf(outfp, "/et g 'Coronal (%.3f cm), %s'\n", cut,
        ctfile);
    fprintf(outfp, "//gt2 'Lateral cross plot (%.2f cm
        axial)'\n", ax1);
    fprintf(outfp, "/et x 'Off Axis Distance (cm)'\n");
    if(maxdose != 0.)
        fprintf(outfp, "/et y 'Normalized Dose'\n");
    else
        fprintf(outfp, "/et y 'Dose (cGy)'\n");
}
else
{
    fprintf(outfp, "//pos 0 0 0.76 1\n");
    fprintf(outfp, "/sm OFF\n");
    fprintf(outfp, "/td z\n");
    fprintf(outfp, "//cx %.2f %.2f\n", la1, la2);
    fprintf(outfp, "//cy %.2f %.2f\n", ax1, ax2);
    fprintf(outfp, "/et g 'Coronal (%.3f cm), %s, %s'\n",
        cut, ctfile, tplan);
    fprintf(outfp, "//gt2 '%i leaves @ %.2f cm width, %.0f
        deg rot incr'\n", lvs, lwid, dginc);
    fprintf(outfp, "//gt3 '%.2f cm GT %.2f cm AB margin,
        %s'\n", gtmargin, abmargin, buffer);
    fprintf(outfp, "/et x 'R <- Lateral (cm) -> L'\n");
    fprintf(outfp, "/et y 'Inf <- Axial (cm) -> Sup'\n");
}
break;
}

// print output
if((iplane == 'A') && (la1 == la2))
{
    for(i = 1; i <= dly; i++)
        fprintf(outfp, "%.4f, %.4e\n", (ap2 - (del * (i - 1))),
            dout[i][1]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else if((iplane == 'A') && (ap1 == ap2))
{
    for(i = 1; i <= dlx; i++)
        fprintf(outfp, "%.4f, %.4e\n", (la1 + (del * (i - 1))),

```

```

        dout[1][i]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else if((iplane == 'S') && (ax1 == ax2))
{
    for(i = 1; i <= dly; i++)
        fprintf(outfp, "%.4f, %.4e\n", (ap2 - (del * (i - 1))),
            dout[i][1]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else if((iplane == 'S') && (ap1 == ap2))
{
    for(i = 1; i <= dlx; i++)
        fprintf(outfp, "%.4f, %.4e\n", (ax1 + (del * (i - 1))),
            dout[1][i]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else if((iplane == 'C') && (la1 == la2))
{
    for(i = 1; i <= dly; i++)
        fprintf(outfp, "%.4f, %.4e\n", (ax2 - (del * (i - 1))),
            dout[i][1]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else if((iplane == 'C') && (ax1 == ax2))
{
    for(i = 1; i <= dlx; i++)
        fprintf(outfp, "%.4f, %.4e\n", (la1 + (del * (i - 1))),
            dout[1][i]);
    if(maxdose != 0.)
        fprintf(outfp, "/or y 0 1.1\n");
}
else
{
    for(i = 1; i <= dly; i++)
    {
        for(j = 1; j <= dlx - 1; j++)
            fprintf(outfp, "%.4e, ", dout[i][j]);
        fprintf(outfp, "%.4e\n", dout[i][dlx]);
    }
}

if(arcs == 1)
{
    // set contour lines to 30%, 50% and 70%
    fprintf(outfp, "//nc\n");
    fprintf(outfp, "/at z 0.30 '.3'\n");
    fprintf(outfp, "/at z 0.70 '.7'\n");
    fprintf(outfp, "/oaas z 0.5 0.6 0\n");
}

```

```

    }
    else
    {
        // set lines for normal SRS contours
        fprintf(outfp, "//nc\n");
        fprintf(outfp, "/at z 0.16 '.16'\n");
        fprintf(outfp, "/at z 0.08 '.08'\n");
        fprintf(outfp, "/oaas z 0 0.4 0\n");
    }
}

printf("%c \n", BACKSPACE);

_setcursortype(_NORMALCURSOR);

fclose(infp);
fclose(outfp);

free(infile);
free(outfile);
free(ctfile);
free(tplan);
free_vector(tang, 0);
free_vector(gst, 0);
free_vector(gsp, 0);
free_vector(wt, 0);
free_ivector(ngp, 0);
free_vector(leaf, 0);
free_matrix(dout, matmin, dly, matmin);
free_matrix(negf, matmin, dly, matmin);
}

// find the dose at the specified dose grid point
float pdose(float fa, float fb, float fg, float ft,
            float of, float aop)
{
    float rval, apx, lax, axx, idf, abf, gtf, idg, abg, gtg, d,
        std, oad, mag, aw, uw, tmr, penum, wd, xep, bfab,
        bfgt, bf, pocr, oar, xinvs, dose;

    // fitting parameters used in OAR
    float alp1 = -0.6373, alp2 = 0.6356;

    // fitting parameters used in build up region TMR
    double tmrfit[] = {0.400, 1.12, -0.689, 0.140};

    // determine the distance from isocenter to dose grid point
    // if rval >= radius of phantom (di/2), then dose = 0
    rval = (float)sqrt((double)(ap * ap + la * la + ax * ax));

    if(rval <= r0)
    {

```

```

// transform BRW (ap,la,ax) into iso BRW [v]p -> xlate to
// isocenter
apx = ap - ap1;
lax = la - lai;
axx = ax - axi;

// transform iso BRW [v]p to fixed machine cs [v]f ->
// rotate table
idf = apx;
abf = lax * tcos + axx * tsin;
gtf = -lax * tsin + axx * tcos;

// transform fixed machine cs [v]f to gantry cs [v]g ->
// rotate gantry
idg = idf * gcos + abf * gsin;
abg = -idf * gsin + abf * gcos;
gtg = gtf;

// determine unknown parameters d, oad, std, aop from
// known parameters:
//      disoi = depth of isocenter
//      idg = distance between isocenter and dose point
// at gantry angle i
//      abg,gtg = off axis position (AB, GT) at gantry
// angle i

// determine depth of dose point: d
d = disoi - idg;

// determine off axis distance: oad
oad = (float)sqrt((double)(abg * abg + gtg * gtg));

// determine primary off center ratio
if(d < dmax)
    pocr = exp((-0.0717 + 0.0650 * d) * oad);
else
{
    if(oad <= 6.)
        pocr = 2. - exp((-0.0055 + 3.0e-4 * d) * oad);
    else
        pocr = 2. - exp((-0.0055 + 3.0e-4 * d) * 6.);
}

// determine source to target distance and magnification
std = sad - idg;
mag = std / scd;

// calculate TMR using fitting function
// if d > dmax -> use exponential fitting
// if d <= dmax -> use polynomial fitting
if(d > dmax)
{

```

```

uw = d - 2.;
aw = -0.05014 + 0.01052 * aop * mag;
tmr = 1.0038 * (float)exp((double)(aw * uw));
}
else
    tmr = (float)poly((double)d, 3, tmrfit);

// calculate OAR using fitting function
penum = ss * (std - scd) / scd;

// find BF(ab)
if(abg < fa * mag)
{
    wd = -1. * (float)fabs((double)((fa * mag) - abg));
    xep = (alp2 * wd) / penum;
    bfab = 0.5 * (float)exp((double)xep);
}

if((abg >= fa * mag) && (abg <= fb * mag))
{
    wd = (float)fabs((double)((fa * mag) - abg));
    xep = (alp1 * wd) / penum;
    bfab = 1. - 0.5 * (float)exp((double)xep);

    wd = (float)fabs((double)((fb * mag) - abg));
    xep = (alp1 * wd) / penum;
    bfab *= 1. - 0.5 * (float)exp((double)xep);
}

if(abg > fb * mag)
{
    wd = -1. * (float)fabs((double)((fb * mag) - abg));
    xep = (alp2 * wd) / penum;
    bfab = 0.5 * (float)exp((double)xep);
}

// find BF(gt)
if(gtg < ft * mag)
{
    wd = -1. * (float)fabs((double)((ft * mag) - gtg));
    xep = (alp2 * wd) / penum;
    bfgt = 0.5 * (float)exp((double)xep);
}

if((gtg >= ft * mag) && (gtg <= fg * mag))
{
    wd = (float)fabs((double)((ft * mag) - gtg));
    xep = (alp1 * wd) / penum;
    bfgt = 1. - 0.5 * (float)exp((double)xep);

    wd = (float)fabs((double)((fg * mag) - gtg));
    xep = (alp1 * wd) / penum;

```

```

    bfgt *= 1. - 0.5 * (float)exp((double)xep);
}

if(gtg > fg * mag)
{
    wd = -1. * (float)fabs((double)((fg * mag) - gtg));
    xep = (alp2 * wd) / penum;
    bfgt = 0.5 * (float)exp((double)xep);
}

// calculate total BF and OAR
bf = bfab * bfgt;
oar = pocr * bf;

// calculate inverse factor
xinvs = (float)pow((double)(sad / std), 2.);

// calculate point dose
dose = of * tmr * oar * xinvs;
}
else // set dose = 0 outside phantom
    dose = 0.;

return dose;
}

// allocate a float matrix with range [rl..rh][cl..ch]
// from "Numerical Recipes in C" [Pre88]
float **matrix(int rl, int rh, int cl, int ch)
{
    int i;
    float **m;

    // allocate pointers to rows
    m = (float **)calloc((unsigned)(rh - rl + 1),
                        sizeof(float *));

    if(m == NULL)
    {
        printf("Allocation failure 1 in matrix()");
        exit(0);
    }
    m -= rl;

    // allocate rows and set pointers to them
    for(i = rl; i <= rh; i++)
    {
        m[i] = (float *)calloc((unsigned)(ch - cl + 1),
                                sizeof(float));

        if(m[i] == NULL)
        {
            printf("Allocation failure 2 in matrix()");
            exit(0);
        }
    }
}

```

```

    }
    m[i] -= cl;
}

// return pointer to array of pointers to rows
return m;
}

// frees a float matrix allocated with matrix()
// from "Numerical Recipes in C" [Pre88]
void free_matrix(float **m, int rl, int rh, int cl)
{
    int i;

    for(i = rh; i >= rl; i--)
        free((char *) (m[i] + cl));
    free((char *) (m + rl));
}

// allocate an integer vector with range [l..h]
// from "Numerical Recipes in C" [Pre88]
int *ivector(int l, int h)
{
    int *v;

    v = (int *)calloc((unsigned)(h - l + 1), sizeof(int));
    if(v == NULL)
    {
        printf("Allocation failure in ivector()");
        exit(0);
    }

    return(v - l);
}

// free an integer vector allocated with vector()
// from "Numerical Recipes in C" [Pre88]
void free_ivector(int *v, int l)
{
    free((char *) (v + l));
}

// allocate a float vector with range [l..h]
// from "Numerical Recipes in C" [Pre88]
float *vector(int l, int h)
{
    float *v;

    v = (float *)calloc((unsigned)(h - l + 1), sizeof(float));
    if(v == NULL)
    {
        printf("Allocation failure in vector()");
    }
}

```

```
    exit(0);
}

return(v - 1);
}

// free a float vector allocated with vector()
// from "Numerical Recipes in C" [Pre88]
void free_vector(float *v, int l)
{
    free((char *)(v + 1));
}
```



APPENDIX D  
DOSE MODEL PROGRAM (VOLUME)

The program LFDVH.C computes a dose volume histogram using as input the output file of LFLOC.C. This process is illustrated in figure D-1. The command line for LFDVH accepts arguments for the input file and for the desired output file name. The output file is suitable for input to ILFN.C, the integrated logistic formula program (discussed in appendix E). This file has the form:

1)	24.856		
2)	64.00	4.99	59.01
3)	0.7324	0.0000	0.7943
4)	0.0275	0.0000	0.0298
5)	0.0210	0.0000	0.0228
6)	0.0234	0.0016	0.0252
7)	0.0209	0.0000	0.0226
8)	0.0211	0.0000	0.0229
9)	0.0207	0.0096	0.0217
10)	0.0200	0.0128	0.0206
11)	0.0316	0.0753	0.0279
12)	0.0814	0.9006	0.0121

Note that the line numbers are not part of the file but are included to facilitate the explanation. Line 1 is a float value denoting the maximum dosimetry value calculated in the volume. The remaining lines are column grouped with the first column representing the total calculation volume, the second column representing the target volume, and the third column representing the remaining volume of normal tissue. The total calculation volume is defined in the

input file and is the product of the extents of the AP, lateral, and axial plane coverages. The histogram dose points in the target are estimated by defining a rectangular prism that conforms to the maximum AP and lateral points in each axial slice in the CT file with a thickness as detailed in the discussion of volume rendering in chapter 4. The target volume is then the total volume multiplied by the quotient of the number of target points and the number of total points calculated in the volume. The normal tissue volume is the difference of the total volume and the target volume. Line 2 contains these values in cubic centimeters in the appropriate columns. Lines 3 through 12 are the histogram bins with line 3 being the 0% to 10% bin, line 4 the 10+% to 20% bin, etcetera. Each bin contains the fraction of points falling in that bin.

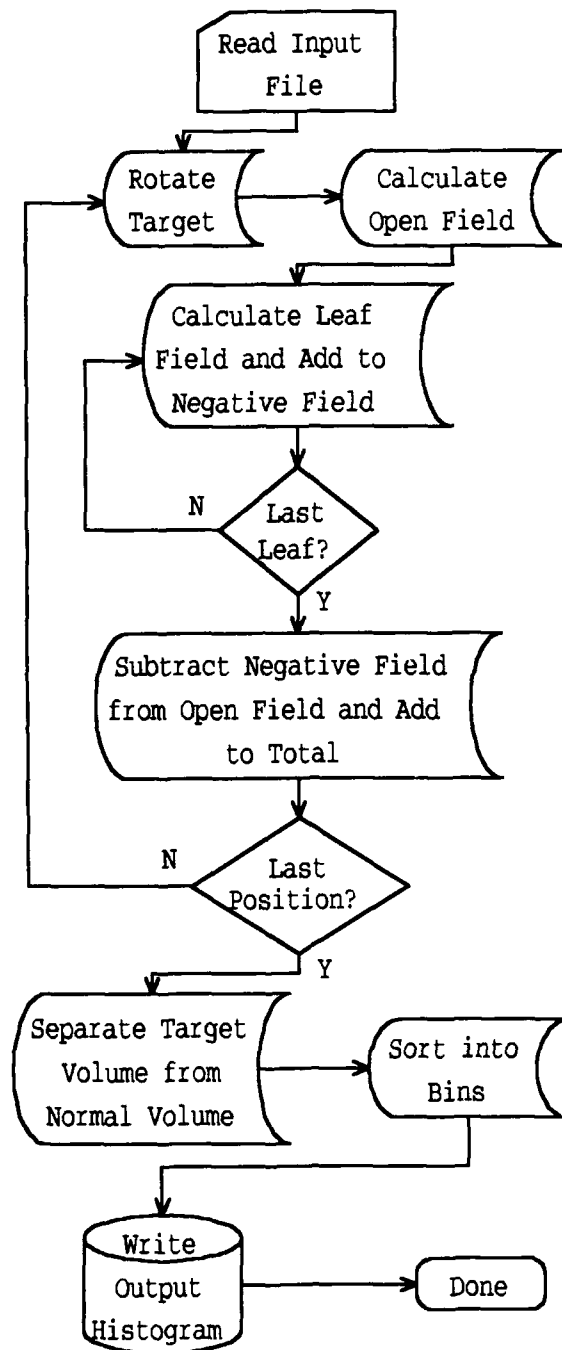


Figure D-1: LFDVH.C flow chart

```

// Program LFDVH.C
//   This program calculates dose volume histograms in 3D
//   space using
//   the negative field model with multileaf collimator
//
// Input is interactive or by command line
// Command line LFDVH <input file> <output file>
//   Input file is from LFLOC.C
//   Output file serves as the input for ILF.C

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <math.h>
#include <conio.h>
#include <string.h>

#define BACKSPACE 8

float pdose(float fa, float fb, float fg, float ft,
            float of, float aop);
float **matrix(int rl, int rh, int cl, int ch, char msg[]);
void free_matrix(float **m, int rl, int rh, int cl);
int *ivector(int l, int h, char msg[]);
void free_ivector(int *v, int l);
float *vector(int l, int h, char msg[]);
void free_vector(float *v, int l);

// set stack size
extern unsigned _stklen = 32767U;

// constant values
//   sad = source to axis distance = 100 cm
//   scd = source to collimator distance = 67.1 cm
//   ss = source size = 0.2 cm
//   dmax = 1.5 cm for 6MVX
static float sad = 100.0,
            scd = 67.1,
            ss = 0.2,
            dmax = 1.5;

// global variables
static float ap, la, ax, api, lai, axi, disoij, di, r0, r2,
            tran, gcos, gsin, tcos, tsin;

void main(int argc, char *argv[])
{
    int i, j, k, arcs, lvs, *ngp, dl, dl1, dl2, dl3, dnum,
        cnum, cside, nocalc, sli, pps, ctpts, tgtpts, spinpos;

    float rad, lwid, ginc, ap0, la0, ax0, ap1, ap2, la1, la2,
        ax1, ax2, del, maxdose, ri2, valc, gang, b, bbcc,

```

```

    lsa, lsb, lsg, lst, *wt, *tang, *gsp, *gst, *leaf,
    **negf, **dout, bfa, bfb, bfg, bft, binsum, tbinsum,
    *bins, *tbins, nbins, nbinsum, thk, *aplim, *lalim,
    *axlim, apmax, apmin, lamax, lamin, tap1, tap2, tla1,
    tla2, tax1, tax2, ctap, ctla, ctax, totvol, tgtvol,
    nrmvol, gtmargin, abmargin, of, mag, caop;

char *infile, *outfile, *ctfile, *tplan,
    spinner[] = {"\\|/-"};

FILE *infp, *outfp, *ctfp;

rad = (M_PI) / 180.;

infile = (char *)malloc(13 * sizeof(char));
outfile = (char *)malloc(13 * sizeof(char));
ctfile = (char *)malloc(13 * sizeof(char));
tplan = (char *)malloc(13 * sizeof(char));

printf("\nMultileaf collimator dose volume histogram\n");

if(argc != 3)
{
    printf("Enter input file name: ");
    scanf("%s", infile);

    printf("Enter output file name: ");
    scanf("%s", outfile);
}
else
{
    infile = argv[1];
    outfile = argv[2];
}

__setcursortype(_NOCURSOR);

if((infp = fopen(infile, "r")) == NULL)
{
    printf("Input file %s unavailable.\n", infile);
    exit(0);
}

if((outfp = fopen(outfile, "w")) == NULL)
{
    printf("Cannot open %s for output.\n", outfile);
    exit(0);
}

printf("Working on %s ...", outfile);

// read CT file and approximate target limits

```

```

fscanf(infp, "%s", ctfile);
if((ctfp = fopen(ctfile, "r")) == NULL)
{
    printf("CT file %s unavailable.\n", ctfile);
    exit(0);
}

fscanf(infp, "%s", tplan);
fscanf(infp, "%f %f", &gtgtmargin, &gtabmargin);

// read CT file
fscanf(ctfp, "%f", >thk);
fscanf(ctfp, "%i", >sli);
fscanf(ctfp, "%i", >pps);

ctpts = sli * pps;
tgtpts = 2 * sli;

aplim = vector(1, tgtpts, "target AP points");
lalim = vector(1, tgtpts, "target lateral points");
axlim = vector(1, tgtpts, "target axial points");

i = 1;
j = 1;
do
{
    apmin = lamin = 20.;
    apmax = lamax = -20.;
    for(k = 1; k <= pps; k++)
    {
        fscanf(ctfp, "%f %f %f", >ctap, >ctla, >ctax);

        if(ctap < apmin)
            apmin = ctap;

        if(ctla < lamin)
            lamin = ctla;

        if(ctap > apmax)
            apmax = ctap;

        if(ctla > lamax)
            lamax = ctla;
    }

    aplim[i] = apmin;
    aplim[i + 1] = apmax;

    lalim[i] = lamin;
    lalim[i + 1] = lamax;

    axlim[i] = ctax + thk / 2.;

```

```

    axlim[i + 1] = ctax - thk / 2.;

    i += 2;
    j += pps;
}while(j <= ctpts);

fclose(ctfp);

// read treatment plan
fscanf(infp, "%i", &arcs);

fscanf(infp, "%f %f %f", &api, &lai, &axi);

fscanf(infp, "%i", &lvs);
fscanf(infp, "%f", &lwid);
bfb = bfg = lvs * lwid / 2.;
bfa = bft = -lvs * lwid / 2.;

fscanf(infp, "%f", &tran);

fscanf(infp, "%f", &ginc);
ginc *= rad;

tang = vector(0, arcs, "table angle");
gst = vector(0, arcs, "gantry start");
gsp = vector(0, arcs, "gantry stop");
wt = vector(0, arcs, "arc weight");
ngp = ivector(0, arcs, "gantry positions");
leaf = vector(0, lvs, "leaves per side");

for(j = 1; j <= arcs; j++)
{
    fscanf(infp, "%f %f %f %f", &tang[j], &gst[j], &gsp[j],
        &wt[j]);
    tang[j] *= rad;
    if(gst[j] == gsp[j])
    {
        gst[j] *= rad;
        gsp[j] = gst[j];
        ngp[j] = 1;
    }
    else
    {
        gst[j] = gst[j] * rad + 0.5 * ginc;
        gsp[j] *= rad;
        ngp[j] = (int)ceil((double)((gsp[j] - gst[j]) / ginc));
    }
}

// read dose grid center
fscanf(infp, "%f %f %f", &ap0, &la0, &ax0);

```

```

// read dose grid size and spacing
fscanf(infp, "%f %f", &ap1, &ap2);
fscanf(infp, "%f %f", &la1, &la2);
fscanf(infp, "%f %f", &ax1, &ax2);
fscanf(infp, "%f", &del);

totvol = (ap2 - ap1) * (la2 - la1) * (ax2 - ax1);
dl1 = (int)ceil((double)((ap2 - ap1) / del));
dl2 = (int)ceil((double)((la2 - la1) / del));
dl3 = (int)ceil((double)((ax2 - ax1) / del));
dl = dl1 * dl2;

printf(" %i dose points to calculate on %i planes\n", dl,
       dl3);
bins = vector(1, 10, "DVH total bins");
tbins = vector(1, 10, "DVH target bins");
dout = matrix(1, dl3, 1, dl, "open field dose");
negf = matrix(1, dl3, 1, dl, "negative field dose");

// begin dose calculation

// patient model di cm diameter sphere
di = 20.;
r0 = 0.5 * di;
r2 = 0.25 * di * di;

spinpos = 0;
for(j = 1; j <= arcs; j++)
{
    printf("\nCalculating arc %i of %i ...\n", j, arcs);

    // parameters used to calculate depth of isocenter, disoi_j
    ri2 = api * api + lai * lai + axi * axi;
    valc = r2 - ri2;

    // calculate dose for gantry angle i (for arc j)
    for(i = 1; i <= ngp[j]; i++)
    {
        printf(" Gantry position %i of %i\r", i, ngp[j]);

        gang = gst[j] + ((float)i - 1.) * ginc;
        gcos = (float)cos((double)gang);
        gsin = (float)sin((double)gang);
        tcos = (float)cos((double)tang[j]);
        tsin = (float)sin((double)tang[j]);

        // calculate depth of isocenter for gantry i and arc j
        b = api * gcos + lai * gsin * tcos + axi * gsin * tsin;
        bbcc = b * b + valc;
        disoi_j = -b + (float)sqrt((double)bbcc);

        // read collimator area/perimeter

```



```

fscanf(infp, "%f", &caop);

// determine output factor for collimator
mag = sad / scd;
of = 1. - 0.4371 * exp(-2.627 * caop * mag);

// calculate point doses for base field
cnum = 0;
for(ax = ax2; ax >= ax1; ax -= del)
{
    cnum++;
    dnum = 0;
    printf("%c%c", spinner[spinpos++], BACKSPACE);
    spinpos &= 0x03;
    for(la = la1; la <= la2; la += del)
        for(ap = ap1; ap <= ap2; ap += del)
        {
            dnum++;
            dout[cnum][dnum] += pdose(bfa, bfb, bfg, bft, of,
                                     caop) * wt[j];
        }
}

// calculate negative field for each collimator leaf
for(cside = 1; cside <= 4; cside++)
{
    for(k = 1; k <= lvs; k++)
        fscanf(infp, "%f", &leaf[k]);

    for(k = 1; k <= lvs; k++)
    {
        nocalc = 0;

        switch(cside)
        {
            // gantry side
            case 1:
                if(leaf[k] >= bfg)
                    nocalc = 1;
                else
                {
                    lsa = bfa + (k - 1) * lwid;
                    lsb = lsa + lwid;
                    lsg = bfg;
                    lst = leaf[k];
                }
                break;

            // B side
            case 2:
                if(leaf[k] >= bfb)
                    nocalc = 1;
        }
    }
}

```

```

else
{
    lsa = leaf[k];
    lsb = bfb;
    lsg = bfg - (k - 1) * lwid;
    lst = lsg - lwid;
}
break;

// table side
case 3:
    if(leaf[k] <= bft)
        nocalc = 1;
    else
    {
        lsa = bfb - k * lwid;
        lsb = lsa + lwid;
        lsg = leaf[k];
        lst = bft;
    }
break;

// A side
case 4:
    if(leaf[k] <= bfa)
        nocalc = 1;
    else
    {
        lsa = bfa;
        lsb = leaf[k];
        lsg = bft + k * lwid;
        lst = lsg - lwid;
    }
break;
}

// calculate negative field if leaf in open base field
if(!nocalc)
{
    // calculate point doses for negative field
    cnum = 0;
    for(ax = ax2; ax >= ax1; ax -= del)
    {
        cnum++;
        dnum = 0;
        printf("%c%c", spinner[spinpos++], BACKSPACE);
        spinpos &= 0x03;
        for(la = la1; la <= la2; la += del)
            for(ap = ap1; ap <= ap2; ap += del)
            {
                dnum++;
                negf[cnum][dnum] += pdose(lsa, lsb, lsg, lst, of,

```

```

                                caop) * wt[j] * (1. - tran);
                                }
                                }
                                } // next leaf k
                                } // next collimator side cside
                                } // next gantry angle i
                                } // next arc j

// subtract negative field from base doses and normalize
printf("\nSubtracting negative fields ...\n");
maxdose = 0.;
cnum = 0;
for(ax = ax2; ax >= ax1; ax -= del)
{
    cnum++;
    dnum = 0;
    for(la = la1; la <= la2; la += del)
        for(ap = ap1; ap <= ap2; ap += del)
        {
            dnum++;

            dout[cnum][dnum] -= negf[cnum][dnum];

            if(dout[cnum][dnum] < tran)
                dout[cnum][dnum] = tran;

            if(dout[cnum][dnum] > maxdose)
                maxdose = dout[cnum][dnum];
        }
}

printf("Sorting into bins ...\n");
binsum = 0.;
tbinsum = 0.;
cnum = 0;
i = 1;
for(ax = ax2; ax >= ax1; ax -= del)
{
    cnum++;
    dnum = 0;
    for(la = la1; la <= la2; la += del)
        for(ap = ap1; ap <= ap2; ap += del)
        {
            dnum++;

            dout[cnum][dnum] /= maxdose;

            if(dout[cnum][dnum] <= 0.1)
                bins[1]++;
            if((dout[cnum][dnum] > 0.1) && (dout[cnum][dnum] <=
                0.2))

```

```

    bins[2]++;
    if((dout[cnum][dnum] > 0.2) && (dout[cnum][dnum] <=
        0.3))
        bins[3]++;
    if((dout[cnum][dnum] > 0.3) && (dout[cnum][dnum] <=
        0.4))
        bins[4]++;
    if((dout[cnum][dnum] > 0.4) && (dout[cnum][dnum] <=
        0.5))
        bins[5]++;
    if((dout[cnum][dnum] > 0.5) && (dout[cnum][dnum] <=
        0.6))
        bins[6]++;
    if((dout[cnum][dnum] > 0.6) && (dout[cnum][dnum] <=
        0.7))
        bins[7]++;
    if((dout[cnum][dnum] > 0.7) && (dout[cnum][dnum] <=
        0.8))
        bins[8]++;
    if((dout[cnum][dnum] > 0.8) && (dout[cnum][dnum] <=
        0.9))
        bins[9]++;
    if(dout[cnum][dnum] > 0.9)
        bins[10]++;

    binsum++;

    if((ax < axlim[i + 1]) && (i <= tgtpts - 3))
        i += 2;

    tax1 = axlim[i + 1];
    tax2 = axlim[i];

    if((ax >= tax1) && (ax <= tax2))
    {
        tla1 = lalim[i];
        tla2 = lalim[i + 1];
        tap1 = aplim[i];
        tap2 = aplim[i + 1];

        if((la >= tla1) && (la <= tla2) && (ap >= tap1) &&
            (ap <= tap2))
        {
            if(dout[cnum][dnum] <= 0.1)
                tbins[1]++;
            if((dout[cnum][dnum] > 0.1) && (dout[cnum][dnum] <=
                0.2))
                tbins[2]++;
            if((dout[cnum][dnum] > 0.2) && (dout[cnum][dnum] <=
                0.3))
                tbins[3]++;
            if((dout[cnum][dnum] > 0.3) && (dout[cnum][dnum] <=

```

```

        0.4))
        tbins[4]++;
        if((dout[cnum][dnum] > 0.4) && (dout[cnum][dnum] <=
            0.5))
            tbins[5]++;
        if((dout[cnum][dnum] > 0.5) && (dout[cnum][dnum] <=
            0.6))
            tbins[6]++;
        if((dout[cnum][dnum] > 0.6) && (dout[cnum][dnum] <=
            0.7))
            tbins[7]++;
        if((dout[cnum][dnum] > 0.7) && (dout[cnum][dnum] <=
            0.8))
            tbins[8]++;
        if((dout[cnum][dnum] > 0.8) && (dout[cnum][dnum] <=
            0.9))
            tbins[9]++;
        if(dout[cnum][dnum] > 0.9)
            tbins[10]++;

        tbinsum++;
    }
}

// print output file
fprintf(outfp, "%.3f\n", maxdose);

tgtvol = totvol * (float)(tbinsum / binsum);
nrmvol = totvol - tgtvol;
fprintf(outfp, "%.2f\t%.2f\t%.2f\n", totvol, tgtvol,
        nrmvol);

nbinsum = binsum - tbinsum;
for(i = 1; i <= 10; i++)
{
    nbins = ((bins[i] - tbins[i]) / nbinsum);
    bins[i] = (bins[i] / binsum);
    tbins[i] = (tbins[i] / tbinsum);
    fprintf(outfp, "%.4f\t%.4f\t%.4f\n", bins[i], tbins[i],
        nbins);
}

printf("%c \n", BACKSPACE);

_setcursortype(_NORMALCURSOR);

fclose(infp);
fclose(outfp);

free_vector(tang, 0);

```

```

free_vector(gst, 0);
free_vector(gsp, 0);
free_vector(wt, 0);
free_ivector(ngp, 0);
free_vector(leaf, 0);
free_vector(bins, 1);
free_vector(tbins, 1);
free_matrix(dout, 1, dl3, 1);
free_matrix(negf, 1, dl3, 1);
free_vector(aplim, 1);
free_vector(lalim, 1);
free_vector(axlim, 1);
}

// find the dose at the specified dose grid point
float pdose(float fa, float fb, float fg, float ft,
            float of, float aop)
{
    float rval, apx, lax, axx, idf, abf, gtf, idg, abg, gtg, d,
          std, oad, aw, uw, tmr, penum, wd, xep, bf, bfab,
          bfgt, oar, xinvs, dose, pocr, mag;

    // fitting parameters used in OAR
    float alp1 = -0.6373, alp2 = 0.6356;

    // fitting parameters used in TMR
    double tmrfit[] = {0.400, 1.12, -0.689, 0.140};

    // determine the distance from origin to dose grid -> rval
    // if rval >= radius of phantom (di/2), then dose = 0
    rval = (float)sqrt((double)(ap * ap + la * la + ax * ax));

    if(rval <= r0)
    {
        // transform BRW (ap,la,ax) into iso BRW [v]p -> xlate to
        // isocenter
        apx = ap - api;
        lax = la - lai;
        axx = ax - axi;

        // transform iso BRW [v]p to fixed machine cs [v]f ->
        // rotate table
        idf = apx;
        abf = lax * tcos + axx * tsin;
        gtf = -lax * tsin + axx * tcos;

        // transform fixed machine cs [v]f to gantry cs [v]g ->
        // rotate gantry
        idg = idf * gcos + abf * gsin;
        abg = -idf * gsin + abf * gcos;
        gtg = gtf;
    }

```

```

// determine unknown parameters: d, r, std, w from known
// parameters:
//      d = depth of dose point
//      idg = distance between isocenter and dose point
// at gantry angle i
//      abg,gtg = off axis position (AB, GT) at gantry
// angle i

// determine depth of dose point: d
d = disoi - idg;

// determine off axis distance: oad
oad = (float)sqrt((double)(abg * abg + gtg * gtg));

// determine primary off center ratio
if(d < dmax)
    pocr = exp((-0.0717 + 0.0650 * d) * oad);
else
{
    if(oad <= 6.)
        pocr = 2. - exp((-0.0055 + 3.0e-4 * d) * oad);
    else
        pocr = 2. - exp((-0.0055 + 3.0e-4 * d) * 6.);
}

// determine source to target distance: std
std = sad - idg;
mag = std / scd;

// calculate TMR using fitting function
//      if d > dmax -> use exponential fitting
//      if d <= dmax -> use polynomial fitting
if(d > dmax)
{
    uw = d - 2.;
    aw = -0.05014 + 0.01052 * aop * mag;
    tmr = 1.0038 * (float)exp((double)(aw * uw));
}
else
    tmr = (float)poly((double)d, 3, tmrfit);

// calculate OAR using fitting function
penum = ss * (std - scd) / scd;

// find BF(ab)
if(abg < fa * mag)
{
    wd = -1. * (float)fabs((double)((fa * mag) - abg));
    xep = (alp2 * wd) / penum;
    bfab = 0.5 * (float)exp((double)xep);
}

```

```

if((abg >= fa * mag) && (abg <= fb * mag))
{
    wd = (float)fabs((double)((fa * mag) - abg));
    xep = (alp1 * wd) / penum;
    bfab = 1. - 0.5 * (float)exp((double)xep);

    wd = (float)fabs((double)((fb * mag) - abg));
    xep = (alp1 * wd) / penum;
    bfab *= 1. - 0.5 * (float)exp((double)xep);
}

if(abg > fb * mag)
{
    wd = -1. * (float)fabs((double)((fb * mag) - abg));
    xep = (alp2 * wd) / penum;
    bfab = 0.5 * (float)exp((double)xep);
}

// find BF(gt)
if(gtg < ft * mag)
{
    wd = -1. * (float)fabs((double)((ft * mag) - gtg));
    xep = (alp2 * wd) / penum;
    bfgt = 0.5 * (float)exp((double)xep);
}

if((gtg >= ft * mag) && (gtg <= fg * mag))
{
    wd = (float)fabs((double)((ft * mag) - gtg));
    xep = (alp1 * wd) / penum;
    bfgt = 1. - 0.5 * (float)exp((double)xep);

    wd = (float)fabs((double)((fg * mag) - gtg));
    xep = (alp1 * wd) / penum;
    bfgt *= 1. - 0.5 * (float)exp((double)xep);
}

if(gtg > fg * mag)
{
    wd = -1. * (float)fabs((double)((fg * mag) - gtg));
    xep = (alp2 * wd) / penum;
    bfgt = 0.5 * (float)exp((double)xep);
}

// calculate total BF and OAR
bf = bfab * bfgt;
oar = pocr * bf;

// calculate inverse factor
xinvs = (float)pow((double)(sad / std), 2.);

// calculate point dose

```



```

    dose = of * tmr * oar * xinvs;
}
else // set dose = 0 outside phantom
    dose = 0.;

return dose;
}

// allocate a float matrix with range [rl..rh][cl..ch]
// from "Numerical Recipes in C" [Pre88]
float **matrix(int rl, int rh, int cl, int ch, char msg[])
{
    int i;
    float **m;

    // allocate pointers to rows
    m = (float **)calloc((unsigned)(rh - rl + 1),
                        sizeof(float *));

    if(m == NULL)
    {
        printf("Allocation failure 1 in matrix (%s)", msg);
        exit(0);
    }
    m -= rl;

    // allocate rows and set pointers to them
    for(i = rl; i <= rh; i++)
    {
        m[i] = (float *)calloc((unsigned)(ch - cl + 1),
                                sizeof(float));

        if(m[i] == NULL)
        {
            printf("Allocation failure 2 in matrix (%s)", msg);
            exit(0);
        }
        m[i] -= cl;
    }

    // return pointer to array of pointers to rows
    return m;
}

// frees a float matrix allocated with matrix()
// from "Numerical Recipes in C" [Pre88]
void free_matrix(float **m, int rl, int rh, int cl)
{
    int i;

    for(i = rh; i >= rl; i--)
        free((char *) (m[i] + cl));
    free((char *) (m + rl));
}

```

```

// allocate an integer vector with range [1..h]
// from "Numerical Recipes in C" [Pre88]
int *ivector(int l, int h, char msg[])
{
    int *v;

    v = (int *)calloc((unsigned)(h - l + 1), sizeof(int));
    if(v == NULL)
    {
        printf("Allocation failure in ivector (%s)", msg);
        exit(0);
    }

    return(v - l);
}

// free an integer vector allocated with vector()
// from "Numerical Recipes in C" [Pre88]
void free_ivector(int *v, int l)
{
    free((char *)(v + l));
}

// allocate a float vector with range [1..h]
// from "Numerical Recipes in C" [Pre88]
float *vector(int l, int h, char msg[])
{
    float *v;

    v = (float *)calloc((unsigned)(h - l + 1), sizeof(float));
    if(v == NULL)
    {
        printf("Allocation failure in vector (%s)", msg);
        exit(0);
    }

    return(v - l);
}

// free a float vector allocated with vector()
// from "Numerical Recipes in C" [Pre88]
void free_vector(float *v, int l)
{
    free((char *)(v + l));
}

```

APPENDIX E  
INTEGRATED LOGISTIC FUNCTION PROGRAM

The program ILFN.C computes the relative complication probability associated with a dose volume histogram using the integrated logistic formula with Neuret weighting as discussed in the papers of Flickinger [Fli89] and Flickinger et al. [Fli90b]. This formula has the form:

$$P(D) = 1 - \prod_i \left( \left[ \frac{NTD_i}{NTD \cdot D_{50}} \right]^k + 1 \right)^{-\frac{v_i}{V}} \quad (E-1)$$

where  $P(D)$  is the probability of complication caused by dose  $D$ ,  $NTD \cdot D_{50}$  is the normalized total dose that results in a 50% complication probability,  $NTD$  is the normalized total dose given to the volume of interest  $v$ ,  $k$  is a statistically determined constant with a recommended value of 12.2, and  $V$  is the total volume.

The factor  $NTD \cdot D_{50}$  is experimentally determined and is dependent on brain volume and, therefore, for strictly comparison purposes (with results only being relative between two or more plans) may be eliminated. Also, for single structures of volume  $v$ , the product may be eliminated, resulting in a simplified model:

$$P(D) = 1 - [NTD^k + 1]^{-\frac{v}{V}} \quad (E-2)$$

The Neuret weighting of the doses is accomplished by evaluating a function for NTD:

$$NTD(Neuret) = \sum_i 0.5206 n_i d_i^2 x_i^{-0.12} \quad (E-3)$$

where  $n$  is the number of fractions (1 for current stereotactic radiosurgery),  $d$  is the dose per fraction in Gy, and  $x$  is the number of fractions per day (also 1 for current stereotactic radiosurgery).

For stereotactic radiosurgery, this function, therefore, reduces to:

$$NTD(Neuret) = \sum_i 0.5206 d_i^2 \quad (E-4)$$

where  $d$  is now the total dose given divided into  $i$  increments each corresponding to the average of a dose volume histogram cell.

Combining equations E-2 and E-4 arrives at the final relative form suitable for stereotactic radiosurgery:

$$P(D) = 1 - \left[ \left( \sum_i 0.5206 d_i^2 \right)^k + 1 \right]^{-\frac{v}{V}} \quad (E-5)$$

which is implemented in ILFN.C.

The command line for this program takes arguments for the input file (a dose volume histogram from LFDVH.C), the desired output file name, the dose to be given to the

volume, and the treatment line percent to which the dose is prescribed. The output file is of the form:

```

1)      T105.DVH ILF P(Neuret)
2)      1000 cGy to the 80 percent line
3)      Total P = 1.627e-02
4)      Target P = 9.796e-01
5)      Normal P = 4.660e-08

```

where the line numbers are not included in the file but are included here to aid explanation. Line 1 specifies the DVH file that was evaluated. Line 2 gives the dose prescription. Lines 3 through 5 enumerate the relative complication probabilities for the total DVH volume, the target volume, and the remaining normal tissue volume, respectively.

As examples of changing variables, we may change the collimator size, the percentage line treated to, and the dose given. Change in collimator size is summarized in table E-1, with dose constant at 1000 cGy and percentage line constant at 80%. As collimator size increases, both the target and the normal tissue ILF increase, as both the target and the normal tissue experience increased dose.

Table E-1: ILF response to collimator size

Collimator (mm)	Target tissue	Normal tissue
12	$3.192 \times 10^{-1}$	$1.164 \times 10^{-9}$
14	$3.982 \times 10^{-1}$	$5.581 \times 10^{-8}$
16	$4.603 \times 10^{-1}$	$2.207 \times 10^{-5}$

Change in percentage line is shown in table E-2, with dose constant at 1000 cGy and collimator size constant at 14

mm. As percentage line increases, both the target and the normal tissue ILF decrease as maximum dose decreases.

Table E-2: ILF response to percentage line

% line	Target tissue	Normal tissue
70	$4.231 \times 10^{-1}$	$1.451 \times 10^{-6}$
80	$3.982 \times 10^{-1}$	$5.581 \times 10^{-8}$
90	$3.753 \times 10^{-1}$	$3.152 \times 10^{-9}$

Change in dose is presented in table E-3, with percentage line constant at 80% and collimator size constant at 14 mm. As dose increases, both the target and the normal tissue ILF increase, as expected.

Table E-3: ILF response to dose

Dose (cGy)	Target tissue	Normal tissue
800	$3.541 \times 10^{-1}$	$2.410 \times 10^{-10}$
1000	$3.982 \times 10^{-1}$	$5.581 \times 10^{-8}$
1200	$4.319 \times 10^{-1}$	$4.772 \times 10^{-6}$

Thus, any of these values may be compared to like values generated from other dose volume histograms, where the lower normal tissue ILF value and the higher target tissue ILF value connotes a better plan based on dose distributed in the volume of interest.

```

// Program ILFN.C
//   Calculates the relative complication probability for an
//   input dose volume histogram using the integrated logistic
//   formula with Neuret weighting [Fli89; Fli90b]
//   Input is either interactive or on the command line
//   Command line ILFN <input file> <output file> <dose>
// <tx line>
//   Input file is from LFDVH.C
//   Dose is in cGy
//   Tx line is a percent

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>
#include <math.h>
#include <dos.h>

float prob(float *dvh, float *doses, float tvol, float vol);
float *vector(int l, int h);
void free_vector(float *v, int l);

int main(int argc, char *argv[])
{
    int i;

    float rxdose, txline, *cavg, *celldose, *totdvh, *tgtdvh,
          *nrmdvh, max, totrcp, tgrcp, nrmrsp, totvol, tgtvol,
          nrmsvol;

    char *infile, *outfile;

    FILE *infp, *outfp;

    infile = (char *)malloc(12 * sizeof(char));
    outfile = (char *)malloc(12 * sizeof(char));

    // compute mean cell range
    cavg = vector(1, 10);
    cavg[1] = .05;
    for(i = 2; i <= 10; i++)
        cavg[i] = cavg[i - 1] + 0.1;

    celldose = vector(1, 10);
    totdvh = vector(1, 10);
    tgtdvh = vector(1, 10);
    nrmdvh = vector(1, 10);

    if(argc != 5)
    {
        printf("\nEnter input file: ");
    }

```

```

scanf("%s", infile);
printf("Enter output file: ");
scanf("%s", outfile);
printf("Enter prescribed dose (cGy): ");
scanf("%f", &rxdose);
printf("Enter treatment line (%): ");
scanf("%f", &txline);
}
else
{
    infile = argv[1];
    outfile = argv[2];
    rxdose = atof(argv[3]);
    txline = atof(argv[4]);
}

if((infp = fopen(infile, "r")) == NULL)
{
    printf("Input file %s unavailable\n", infile);
    exit(0);
}

if((outfp = fopen(outfile, "w")) == NULL)
{
    printf("Output file %s cannot be opened\n", outfile);
    exit(0);
}

// calculate average cell doses
for(i = 1; i <= 10; i++)
    celldose[i] = rxdose * (cavg[i] / (txline / 100.));

// read input DVH
fscanf(infp, "%f", &max);
fscanf(infp, "%f %f %f", &totvol, &tgtvol, &nrmvol);

for(i = 1; i <= 10; i++)
    fscanf(infp, "%f %f %f", &totdvh[i], &tgtdvh[i],
        &nrmdvh[i]);

for(i = 1; i <= 10; i++)
{
    totdvh[i] /= 100.;
    tgtdvh[i] /= 100.;
    nrmdvh[i] /= 100.;
}

fclose(infp);

// calculate complication probabilities
totrcp = prob(totdvh, celldose, totvol, totvol);
tgtrcp = prob(tgt dvh, celldose, totvol, tgtvol);

```



```

nrmrcp = prob(nrmdvh, celldose, totdvol, nrmvol);

printf("%s ILF P(Neuret)\n", strupr(infile));
printf("%.0f cGy to the %.0f percent line\n",
        rxdose, txline);
printf("  Total  P = %.3e\n", totrcp);
printf("  Target P = %.3e\n", tgtrcp);
printf("  Normal P = %.3e\n", nrmrcp);

fprintf(outfp, "%s ILF P(Neuret)\n", strupr(infile));
fprintf(outfp, "%.0f cGy to the %.0f percent line\n",
        rxdose, txline);
fprintf(outfp, "  Total  P = %.3e\n", totrcp);
fprintf(outfp, "  Target P = %.3e\n", tgtrcp);
fprintf(outfp, "  Normal P = %.3e\n", nrmrcp);

free(infile);
free(outfile);
free_vector(cavg, 1);
free_vector(celldose, 1);
free_vector(totdvh, 1);
free_vector(tgtdvh, 1);
free_vector(nrmdvh, 1);

return(1);
}

// compute complication probability
float prob(float *dvh, float *doses, float tvol, float vol)
{
    int i;

    float csum, cdose, p, viv, inner;

    viv = -1. * (vol / tvol);

    csum = 0.;
    for(i = 1; i <= 10; i++)
    {
        cdose = dvh[i] * doses[i];

        csum += 0.5206 * pow(cdose, 2.);
    }

    inner = pow(csum, 12.2);

    p = 1. - pow((1. + inner), viv);

    return p;
}

```

```
// allocates a float vector with range [l..h]
// from "Numerical Recipes in C" [Pre88]
float *vector(int l, int h)
{
    float *v;

    v = (float *)calloc((unsigned)(h - l + 1), sizeof(float));
    if(v == NULL)
    {
        printf("Allocation failure in vector()\n");
        exit(0);
    }

    return(v - l);
}

// frees a vector allocated with vector
// from "Numerical Recipes in C" [Pre88]
void free_vector(float *v, int l)
{
    free((char *)(v + l));
}
```

## REFERENCES

- Arc85 G. Arcovito, A. Piermattei, G. D'Abramo, F.A. Bassi, "Dose measurements and calculations of small radiation fields for 9-MV x rays," Med. Phys. 12:779 (1985)
- Bet84 O.O. Betti, V.E. Derechinsky, "Hyperselective encephalic irradiation with linear accelerator," Acta Neurochirurgica Suppl. 33:385 (1984)
- Bja82 B.E. Bjarngard, L.H. Brown, G.K. Svensson, "Scatter dose decrement values for rectangular fields," Med. Phys. 9:830 (1982)
- Bja90 B.E. Bjarngard, J.S. Tsai, R.K. Rice, "Doses on the central axes of narrow 6-MV x-ray beams," Med. Phys. 17:794 (1990)
- Boe90 R. Boesecke, T. Bruckner, G. Ende, "Landmark based correlation of medical images," Phys. Med. Biol. 35:121 (1990)
- Bor90 T. Bortfeld, J. Burkelbach, R. Boesecke, W. Schlegel, "Methods of image reconstruction from projections applied to conformation radiotherapy," Phys. Med. Biol. 35:1423 (1990)
- Bov90 F.J. Bova, "Radiation Physics," Neurosurgery Clinics of North America 1:909 (1990)
- Bov91 F.J. Bova, W.J. Friedman, "Stereotactic angiography: An inadequate database for radiosurgery?" Int. J. Radiation Oncology Biol. Phys. 20:891 (1991)
- Boy85 A. Boyer, E. Mok, "A photon dose distribution model employing convolution calculations," Med. Phys. 12:169 (1985)
- Boy86 A.L. Boyer, E.C. Mok, "Calculation of photon dose distributions in an inhomogeneous medium using convolutions," Med. Phys. 13:503 (1986)

- Chi81 L.M. Chin, P.K. Kijewski, G.K. Svensson, J.T. Chaffey, M.B. Levene, B.E. Bjarngard, "A computer-controlled radiation therapy machine for pelvic and para-aortic nodal areas," *Int. J. Radiation Oncology Biol. Phys.* 7:61 (1981)
- Chu86 C.S. Chui, R. Mohan, "Off-center ratios for three-dimensional dose calculations," *Med. Phys.* 13:409 (1986)
- Chu88a C.S. Chui, R. Mohan, D. Fontanella, "Dose computations for asymmetric fields defined by independent jaws," *Med. Phys.* 15:92 (1988)
- Chu88b C.S. Chui, R. Mohan, "Extraction of pencil beam kernels by the deconvolution method," *Med. Phys.* 15:138 (1988)
- Fit75 L.T. Fitzgerald, W. Mauderli, "Analysis of errors in three-dimensional reconstruction of radium implants from stereo radiographs," *Radiology* 115:455 (1975)
- Fli89 J.C. Flickinger, "An integrated logistic formula for prediction of complications from radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 12:879 (1989)
- Fli90a J.C. Flickinger, L.D. Lunsford, A. Wu, A.H. Maitz, A.M. Kalend, "Treatment planning for gamma knife radiosurgery with multiple isocenters," *Int. J. Radiation Oncology Biol. Phys.* 18:1495 (1990)
- Fli90b J.C. Flickinger, M.C. Schell, D.A. Larson, "Estimation of complications for linear accelerator radiosurgery with the integrated logistic formula," *Int. J. Radiation Oncology Biol. Phys.* 19:143 (1990)
- Fli90c J.C. Flickinger, A. Maitz, A. Kalend, L.D. Lunsford, A. Wu, "Treatment volume shaping with selective beam blocking using the Leksell gamma unit," *Int. J. Radiation Oncology Biol. Phys.* 19:783 (1990)
- Fol82 J.D. Foley, A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading MA (1984)
- Fri89a W.A. Friedman, F.J. Bova, "Stereotactic radiosurgery," *Contemporary Neurosurgery* 11:1 (1989)
- Fri89b W.A. Friedman, F.J. Bova, "The University of Florida radiosurgery system," *Surg. Neurol.* 32:334 (1989)

- Fri90 W.A. Friedman, "LINAC radiosurgery," *Neurosurgery Clinics of North America* 1:991 (1990)
- Gil90 P.L. Gildenberg, "The history of stereotactic neurosurgery," *Neurosurgery Clinics of North America* 1:765 (1990)
- Har85 G.H. Hartmann, W. Schlegel, V. Sturm, B. Kober, C. Pastyr, W.J. Lorenz, "Cerebral radiation surgery using moving field irradiation at a linear accelerator facility," *Int. J. Radiation Oncology Biol. Phys.* 11:1185 (1985)
- Iwa85 A. Iwasaki, "A method of calculating high-energy photon primary absorbed dose in water using forward and backward spread dose-distribution functions," *Med. Phys.* 12:731 (1985)
- Ker80 J.G. Kereiakes, M. Rosenstein, CRC Handbook of Radiation Doses in Nuclear Medicine and Diagnostic X-Ray, CRC Press, Boca Raton FL (1980)
- Kha70 F.M. Khan, "Computer dosimetry of partially blocked fields in cobalt teletherapy," *Radiology* 97:405 (1970)
- Kha80 F.M. Khan, W. Sewchand, J. Lee, J.F. Williamson, "Revision of tissue-maximum ratio and scatter-maximum ratio concepts for cobalt 60 and higher energy x-ray beams," *Med. Phys.* 7:230 (1980)
- Kha84 F.M. Khan, The Physics of Radiation Therapy, Williams and Wilkins, Baltimore MD (1984)
- Lar90 D.A. Larson, P.H. Gutin, "Introduction to radiosurgery," *Neurosurgery Clinics of North America* 1:897 (1990)
- Lea87 D.D. Leavitt, J.R. Stewart, "Optimization of electron arc therapy doses by dynamic collimator control," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lea89 D.D. Leavitt, J.R. Stewart, J.H. Moeller, W.L. Lee, G.A. Takach Jr., "Electron arc therapy: Design, implementation and evaluation of a dynamic multi-vane collimator system," *Int. J. Radiation Oncology Biol. Phys.* 17:1089 (1989)

- Lea91 D.D. Leavitt, F.A. Gibbs Jr., M.P. Heilbrun, J.H. Moeller, G.A. Takach Jr., "Dynamic field shaping to optimize stereotactic radiosurgery," *Int. J. Radiation Oncology Biol. Phys.* 21:1247 (1991)
- Lek51 I. Leksell, "The stereotaxic method and radiosurgery of the brain," *Acta Chir. Scand.* 102:316 (1951)
- Low90 N.N. Low, S. Vijayakumar, I. Rosenberg, S. Rubin, R. Virudachalam, D.R. Spelbring, G.T.Y. Chen, "Beam's eye view based prostate treatment planning: Is it useful?" *Int. J. Radiation Oncology Biol. Phys.* 19:759 (1990)
- Lul87 B. Lulu, "Conversion of CT coordinates to stereotaxic frame coordinates for the Brown-Roberts-Wells frame," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Lut88 W. Lutz, K.R. Winston, N. Maleki, "A system for stereotactic radiosurgery with a linear accelerator," *Int. J. Radiation Oncology Biol. Phys.* 14:373 (1988)
- Man77 J. Mantel, H. Perry, "Automatic variation of field size and dose rate in rotation therapy," *Int. J. Radiation Oncology Biol. Phys.* 2:697 (1977)
- Met89 C.E. Metz, L.E. Fencil, "Determination of three-dimensional structure in biplane radiography without prior knowledge of the relationship between the two views: Theory," *Med. Phys.* 16:45 (1989)
- Moh87 R. Mohan, C.S. Chui, "Use of fast Fourier transforms in calculating dose distributions for irregularly shaped fields for three dimensional treatment planning," *Med. Phys.* 14:70 (1987)
- Moh88 R. Mohan, G. Barest, L.J. Brewster, C.S. Chui, G.J. Kutcher, J.S. Laughlin, Z. Fuks, "A comprehensive three-dimensional radiation treatment planning system," *Int. J. Radiation Oncology Biol. Phys.* 15:481 (1988)
- Myr88 L.C. Myriantopoulos, G.T.Y. Chen, D.R. Spelbring, C.A. Pelizzari, I. Rosenberg, S. Vijayakumar, H.J. Halpern, M. Reese, "Beam's eye view volumetrics: An aid in rapid treatment plan optimization," 30th Annual Meeting of the American Society for Therapeutic Radiology and Oncology, Manuscript (1988)

- Phi91 Philips Medical Systems, Stereotactic Radiosurgery Operations Manual, Release 1.12.12, Philips Medical Systems U.K. (1991)
- Pik87 B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, "Dose distributions in dynamic stereotactic radiosurgery," *Med. Phys.* 14:780 (1987)
- Pik90 G.B. Pike, E.B. Podgorsak, T.M. Peters, C. Pla, A. Olivier, L. Souhami, "Dose distributions in radiosurgery," *Med. Phys.* 7:296 (1990)
- Pre88 W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge U.K. (1988)
- Ric87a R.K. Rice, J.L. Hansen, G.K. Svensson, R.L. Siddon, "Measurement of dose distributions in small beams of 6 MV x-rays," *Phys. Med. Biol.* 32:1087 (1987)
- Ric87b J. Richter, P. Klemm, M. Neumann, C. Nowak, O. Sauer, "Computer control of a linac for dynamic treatment," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Saw87a C.B. Saw, K. Ayyangar, N. Suntharalingam, "Coordinate transformations and calculation of the angular and depth parameters for a stereotactic system," *Med. Phys.* 14:1042 (1987)
- Saw87b C.B. Saw, K. Ayyangar, N. Suntharalingam, "A simple algorithm for calculating the angular coordinates and depth for the BRW stereotaxic system," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sid86 R.L. Siddon, "3-D treatment planning algorithms," 1986 American Association of Physicists in Medicine Summer School Proceedings, 533 (1986)
- Sid87 R.L. Siddon, N.H. Barth, "Stereotaxic localization of intracranial targets," *Int. J. Radiation Oncology Biol. Phys.* 13:1241 (1987)
- Sou90 L. Souhami, A. Olivier, E.B. Podgorsak, M. Pla, G.B. Pike, "Radiosurgery of cerebral arteriovenous malformations with the dynamic stereotactic irradiation," *Int. J. Radiation Oncology Biol. Phys.* 19:775 (1990)

- Spe87 D.R. Spelbring, G.T.Y. Chen, C.A. Pelizarri, A.Awan, H. Sutton, "A computer simulated evaluation of multi-vane collimator beam delivery systems," The Use of Computers in Radiation Therapy, I.A.D. Bruinvis, P.H. van der Giessen, H.J. van Kleffens, F.W. Wittkämper (Editors), Elsevier Science Publishers B.V., Holland (1987)
- Sta88 G. Starkschall, "A convolution method for constructing primary beam profiles in the presence of beam modifiers," Med. Phys. 15:657 (1988)
- Suh90 T.S. Suh, "Optimization of dose distribution for the system of linear accelerator based stereotactic radiosurgery," Ph.D. Dissertation, University of Florida (1990)
- Toe90 K.D. Toennies, J.K. Udupa, G.T. Herman, I.L. Wornam, S.R. Buchman, "Registration of 3D objects and surfaces," IEEE Computer Graphics and Applications 10:52 (1990)
- Win88 K.R. Winston, W. Lutz, "Linear accelerator as a neurosurgical tool for stereotactic radiosurgery," Neurosurgery 22:454 (1988)



## BIOGRAPHICAL SKETCH

Dale C. Moss was born on 30 July 1952 in Los Angeles, California. He later moved with his family to Honolulu, Hawaii, where he graduated from McKinley High School in 1970. He served in the U.S. Navy as a fire control radar technician and participated in three deployments to the Tonkin Gulf during the Viet Nam conflict. He then returned to Honolulu, entered the University of Hawaii at Manoa, and graduated with a Bachelor of Science degree in electrical engineering in 1980. After working for the U.S. Navy as a civilian engineer at the Pacific Missile Test Center in electromagnetics research, he entered the U.S. Air Force Medical Service as a Bioenvironmental Engineer. His first duty station was Vandenberg Air Force Base, where he worked in military public health and environmental protection on the Space Shuttle project. He was then selected for an Air Force sponsored graduate program at the University of Florida, resulting in a Master of Science degree in medical health physics in 1986. He was next assigned to the Air Force Regional Medical Center, Clark Air Base, Philippines, as base and medical center radiation protection officer and as the deputy base bioenvironmental engineer. He was later transferred to the Air Force Occupational and Environmental

Health Laboratory operating location at Clark Air Base as chief of radiation services, with health and medical physics consulting responsibility for Air Force and Navy units in the western Pacific. He was again selected for an Air Force sponsored graduate program, and re-entered the University of Florida in August of 1989 in the medical physics Ph.D. program, where he has been engaged in clinical work and has assisted in projects involved with the University of Florida Stereotactic Radiosurgery System.

He is a member of the Air Force Association, a member of the Marine Corps Association, a member of ANΣ (the national honor society for nuclear science and engineering), and a student member of the American Association of Physicists in Medicine.